

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a kybernetiky
Katedra kybernetiky a biomedicínského inženýrství

DIPLOMOVÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a kybernetiky
Katedra kybernetiky a biomedicínského inženýrství

**Mobilní aplikace pro analýzu zvukových vstupů mobilního
zařízení za účelem detekce epileptických záchvatů**
**Mobile Application for Input Source Analysis of Mobil
Device for Epileptic State Detection of Children**

Zadání diplomové práce

Student: **Bc. David Vybíral**

Studijní program: N2649 Elektrotechnika

Studijní obor: 3901T009 Biomedicínské inženýrství

Téma: Mobilní aplikace pro analýzu zvukových vstupů mobilního zařízení
za účelem detekce epileptických záchvatů dětí
Mobile Application for Input Source Analysis of Mobil Device
for Epileptic State Detection of Children

Zásady pro vypracování:

Současná mobilní PDA zařízení disponují rychlými procesory, které umožňují provozovat i velmi náročné aplikace typu analýzy a zpracování vstupního signálu z mikrofону či akcelerometru. Cílem práce je vyvinout aplikaci pro mobilní zařízení PDA/SmartPhone, která bude zpracovávat měřený signál (v závislosti na rychlosti možnostech HW v reálném čase) a provádět na něm FFT a CWT analýzu. V závislosti na kvalitě výsledné informace může být použita i fuzzy klasifikace, jejíž volba bude zdůvodněna.

Práce bude obsahovat tyto body:

1. Zpracování teoretické problematiky zpracování audio signálu na platformě Android (Java + ADK).
2. Zpracování teoretické problematiky použití FFT a CWT analýzy.
3. Návrh a realizace aplikace pro mobilní zařízení pro analýzu externího signálu.
4. Implementace vhodných algoritmů pro detekce epileptických záchvatů.
5. Otestování vyvinuté aplikace na reálných vstupních datech získaných na vzorku pacientů (ve spolupráci s FN Ostrava).
6. Diskuze dosažených výsledků.

Seznam doporučené odborné literatury:


1. MURPHY, Mark, L. *The Busy Coder's Guide to Advanced Android Development*. (USA) : CommonsWare,LLC., 2008-2009. 748 s. ISBN 978-0-9816780-0-9.
2. MEIER, Retto. *Professional Android 2 Application Development*. 2nd Edition. Wrox, 2010. 576 s. ISBN 978-0470565520.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Ondřej Krejcar, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.


.....

Bc. David Vybírál

V Ostravě, dne 30. dubna 2012

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Ondřeji Krejcarovi, Ph.D. a Ing. Jakubovi Jirkovi za účinnou pedagogickou a odbornou pomoc a další cenné rady v průběhu zpracovávání práce.

Abstrakt

Tato diplomová práce se zabývá programováním aplikace pro operační systém Android, jejímž úkolem je detekce probíhajícího epileptického záchvatu určitého typu. Za tímto účelem byl navržen algoritmus. Princip samotného vyhodnocení zvukových nálezů je realizován pomocí porovnávání energií a vzájemných vzdáleností nálezů a posléze srovnávání dominantní frekvence z frekvenčního spektra nálezů vyhodnocených algoritmem jako anomálie. Vyvinutá aplikace byla otestována na datech získaných ve spolupráci s FN Ostrava.

Klíčové slova

Epilepsie, Android, Dalvik, Eclipse

Abstract

This thesis deals with the programming of application for Android operating system, The application task is to detect ongoing epileptic seizure of specific type. For this purpose, algorithm was designed. The principle of sound evaluation findings is realized by comparing the energy and the mutual distance of the findings and then comparing the dominant frequencies of the frequency spectrum findings evaluated by algorithm as an anomaly. Developed application was tested on data obtained in collaboration with University hospital Ostrava.

Keywords

Epilepsie, Android, Dalvik, Eclipse

Obsah

1	Úvod	8
2	Epilepsie obecně	9
2.1	Historie	9
2.2	Výskyt epilepsie	10
2.3	Rozvoj epilepsie	11
2.3.1	3 stádia epileptického záchvatu	12
2.4	Klasifikace	12
2.4.1	Generalizované záchvaty	13
2.4.2	Parciální ložiskové záchvaty	14
2.4.3	Status epilepticus	14
2.5	Identifikace na základě zvukových projevů	14
3	Android	15
3.1	Historie OS Android	15
3.2	Android a jeho konkurenti	16
3.3	Architektura	17
3.4	Virtuální stroj Dalvik VM	18
3.5	Tvorba aplikací	19
3.5.1	Android SDK	19
3.5.2	Vývojové prostředí Eclipse	19
3.6	Základní části aplikace	21
3.6.1	Activity	21
3.6.2	Service	22
3.6.3	Content provider	22
3.6.4	Broadcast receiver	22
4	Návrh grafického uživatelského rozhraní	23
4.1	Prvky GUI	23
4.1.1	TextView a Edit Text	23
4.1.2	Button	23
4.2	Rozvržení prvků	24
4.2.1	LinearLayout	24
4.2.2	RelativeLayout	25
4.2.3	ListView	25
4.2.4	Ostatní layouty	25
5	Zpracování zvuku	26
5.1	Fourierova transformace	27
5.1.1	Fourierova transformace diskrétních signálů	27
5.1.2	Diskrétní Fourierova transformace	27
5.1.3	Rychlá Fourierova transformace	28
5.2	Vlnková transformace	29
5.2.1	Wavelet-vlnka	29
5.2.2	Spojité vlnkové transformace	29
5.2.3	Diskrétní vlnkové transformace	30
6	Návrh aplikace pro analýzu externího signálu	31
6.1	Nahrávání zvuku	31

6.2	Rozpoznání zvukového nálezu	33
6.3	Informace o nálezu	35
6.4	Algoritmus pro detekci epileptických záchvatů	38
6.5	Vzhled aplikace	40
7	Testování	42
7.1	Klidový stav pacienta	42
7.2	Epileptická aktivita pacienta	43
7.3	Diskuze	45
8	Závěr	46
9	Literatura	47

1 Úvod

Operační systém Android je velice mladou platformou, která se stále vyvíjí. Vzhledem k tomu, že je licencován jako opensource má každý jeho uživatel možnost z internetu bezplatně stáhnout vývojový balíček a pustit se do tvorby své vlastní aplikace.

Motivací ke vzniku této práce bylo prozkoumat schopnosti mobilních telefonů vybavených Androidem a za pomoci volně šiřitelných vývojových nástrojů realizovat aplikaci, pro detekci vybraných specifických skupin epileptických záchvatu na základě zvukových projevů pacientů.

Během tvorby práce byly postupně sbírány poznatky, které jsou shrnuty v následujících kapitolách. Teoretická část práce je věnována popisu epilepsie počínaje historií, výskytem až po klasifikaci. Druhá část teoretické části se zabývá problematikou OS android a tvorbou aplikací na tuto platformu. Dále je práce navázána praktickou částí, ve které je názorně zdokumentován postup vývoje aplikace a její funkce, završena testováním.

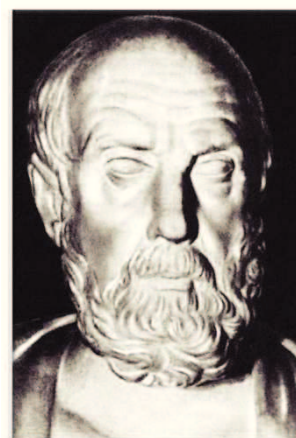
2 Epilepsie obecně

Epilepsie je nemoc, která pacienta velmi obtěžuje a limituje v řadě aktivit. Projevuje se chorobným stavem, který vzniká nadměrnou elektrickou aktivitou v mozku a nervovém systému. Nervové buňky mezi sebou komunikují pomocí elektrických signálů. Za normálních okolností přenášejí elektrické vzruchy dokonale regulovaným systémem. Nemocní epilepsií však zažívají okamžiky, kdy velké množství nervových buněk je aktivováno najednou. Tento nekontrolovaný výboj má širokou paletu příznaků - od pouhého prázdného pohledu až po ztrátu vědomí provázenou křečemi. Tyto stavy se nazývají epileptické záchvaty. Ojedinělý epileptický záchvat ještě nemusí znamenat epilepsii.

[1]

2.1 Historie

Název epilepsie je odvozen z řeckého slova epilambanein, které znamená zachvacovat. První, i když ne docela spolehlivý písemný údaj o epilepsii byl objeven v babylonském Chamurabiho zákoníku z r. 1900 před n.l.. Ve starověku epilepsii označovali jako morbus sacer nebo morbus divinus. Tato označení jsou svědectvím, že se epileptické záchvaty přičítaly nadpřirozeným jevům. Až do Hippokrata byla epilepsie v rukou kněží. Hippokrates pokládal epilepsii za dědičnou nemoc. Pravděpodobně jako první odlišil od celkových křečí částečné, jacksonské křeče.



Obrázek 1.: Hippokrates

Teprve Tissot (1770) činí nový pokus o rozdělení epilepsií. Rozeznává 3 druhy epilepsie: idiopatickou, u níž lze zjistit anatomickou příčinu v mozku, esenciální, u níž příčinu zjistit nelze a sympatetickou, u níž je původce epilepsie mimo ústřední nervovou soustavu.

Fritsch a Hitzig (1870) konají první pokusy o vyvolání záchvatu lokálním podrážděním mozkové kůry elektrickým proudem. V tom samém roce uveřejňuje Hughlings Jackson práci, v níž uvádí, že záchvaty vznikají výbojem ložiska v ústřední nervové soustavě, odkud se šíří podráždění na větší či menší oblast. Epilepsii definuje jako občasný, náhlý, nadměrný, místní výboj šedé hmoty. Hlavní úlohu hraje omezená oblast šedé hmoty, která se stává z různých příčin instabilní, čili pohotová k výboji. Každá epilepsie je fokální a podle sídla ložiska nízkotažová, středně-etážová a vysoko-etážová. Vlivem této teorie se začíná soustřeďovat pozornost k epileptogennímu ložisku.

Významným objevem, který přispěl k rozvoji bádání o epilepsii, bylo zavedení elektroencefalografie EEG do klinické praxe. Již v roce 1875 pozoroval Caton, že mozek při své činnosti vytváří nesmírně slabé elektrické proudy. Tuto elektrickou činnost mozku u člověka zachytil v roce 1928 H.Berger a nazval ji elektroencefalogramem.

[2]

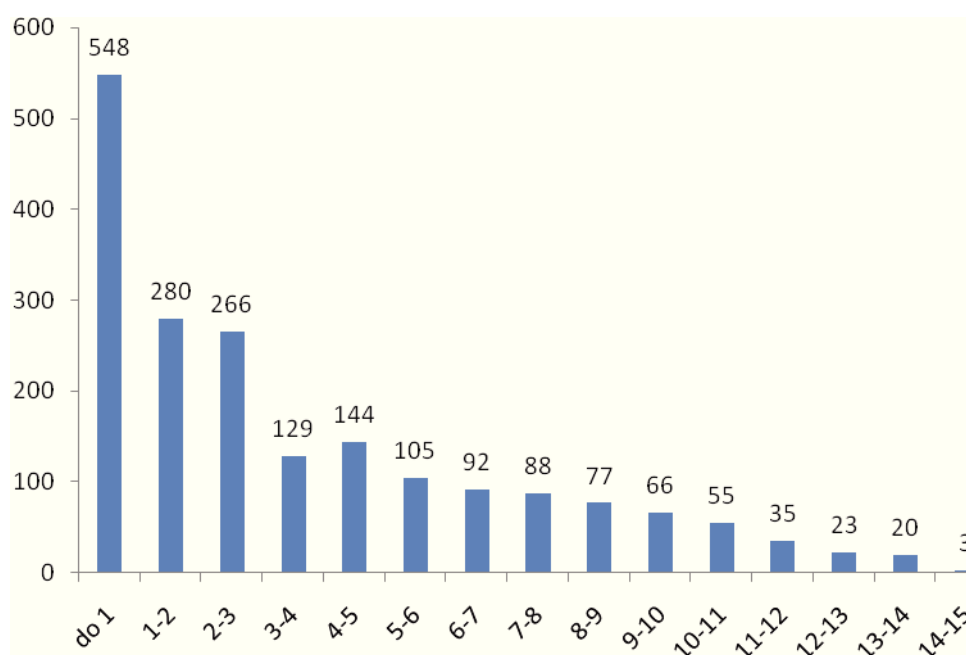
2.2 Výskyt epilepsie

Výskyt epilepsie je zaznamenán na celém světě bez ohledu na věk, rasu a zeměpisnou polohu. V rozvinutých zemích je incidence nových případů epilepsie 24-53 na 100 tisíc osob za rok. V méně rozvinutých zemích Jižní Ameriky je uváděna incidence 110-190 na 100 tisíc osob za rok. Ze statistik vyplývá, že největší incidence je u dětí, především v prvních měsících života. U dospělých osob incidence klesá a ve stáří opět roste. Statistický odhad dále ukazuje, že asi 1-1,5% populace trpí epilepsií. Pravděpodobně více než 50% nemocných má parciální záchvaty.

[3]

V minulém století byla na dvou významných dětských klinikách získána data o pacientech. Jednalo se o dětskou neurologickou ambulanci středočeského kraje, kterou prošlo v období od r. 1953 do konce r. 1960 celkem 872 dětí s nervovými záchvaty a klinickou ambulancí II. dětské kliniky, kterou od r. 1945 do konce r. 1960 prošlo celkem 2267 dětí. Na epileptické záchvaty připadalo z celkového počtu 3139 dětí celkem z nich 2153. Percentuální vyčíslení na dětskou populaci nešlo provést, protože ambulanci a klinikou neprošly všechny děti z jednoho kraje se záchvatovými onemocněními. Nicméně z 2153 nemocných dětských pacientů, byl počátek záchvatu přesně zjištěn u 1931 z nich. Graf sestaven z těchto dat jednoznačně ukazuje, že začátek onemocnění spadá většinou do prvních dvou let života, během dalších let výskyt plynule klesá.

[2]

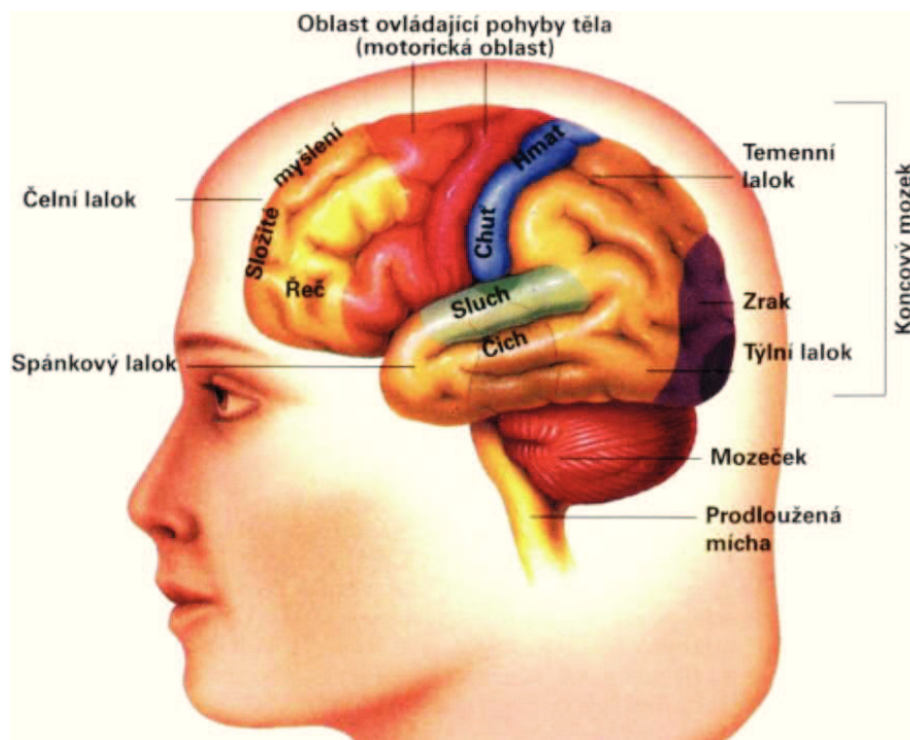


Obrázek 2.: Začátek záchvatu u 1931 dětí ve věku do 15 let

2.3 Rozvoj epilepsie

Hlavním projevem epilepsie je epileptický záchvat, což je proces narušující funkční hierarchii mozku, vnucováním normálně fungujícím strukturám a komplexům abnormální aktivitu. Důsledkem toho vzniká porucha funkcí, které postižené místo za normálních okolností zajišťují, zhoršování vyšší nervové činnosti, paměti, motorické koordinace apod. v závislosti na lokalizaci epileptického ohniska či ohnisek.

[3]



Obrázek 3.: Mozková centra

Základem celého procesu je vznik epileptického agregátu inzultem (vyvolávající příčina chorobného stavu), po jehož zásahu dochází ke vzniku určitého množství epileptických neuronů cestou neurotoxické kaskády, které pak mají z funkčního pohledu změněnou reaktivitu na chemické, mechanické nebo elektrické podněty. Ke klinické manifestaci je potřebné určité kritické množství epileptických neuronů pracujících synchronně.

Pro rozvoj samotného záchvatu, je důležitých několik faktorů. **Záchvatovou pohotovost** můžeme charakterizovat jako náchylnost mozkové tkáně k hyperexcitabilitě a hypersynchronizaci, projevující se zejména v určitých zátěžových podmínkách. Mění se s věkem a stavem vnitřního prostředí a je geneticky podmíněna. **Ohnisko** se může vytvořit kdekoli v mozku působením inzultu. **Podnět**, který není bezpodmínečně přítomen (světlo, hluk, změna metabolismu apod.). Za **spouštěcí oblast** pokládáme nejlabilnější oblast s vysokou instabilitou a schopnosti spontánního vzniku epileptických výbojů, ta ne vždy odpovídá místě vzniku epileptického ohniska. **Stavidla** jsou místa v mozku, která mohou zabránit rozvoji šíření a tím pádem generalizaci záchvatu.

[3]

2.3.1 3 stádia epileptického záchvatu:

Období před záchvatem

Prodromální fáze (trvá několik dní až hodin) a aura (několik sekund). Obě fáze mají stejné projevy, těžko se odliší, nastávají změny v chování – dezorientace, apatie, změny v pohybu, pocity strachu, možné i zvracení atd.. Někdy tato fáze chybí nebo je těžko rozeznatelná.

Epileptický záchvat

Může se objevit v níže popsaných druzích, při ztrátě vědomí se již jedná o vážný stav. Trvá většinou několik minut, při překročení 20 minut je potřebné provést speciální zákrok.

Období po záchvatu

Postupný návrat do bezzáchvatového stavu, trvá 10-30 minut, některé příznaky jsou shodné s aurou - dezorientace, potácivá a nekoordinovaná chůze, apatie, pocity strachu až agrese, může se objevit mžourání až dočasná slepota, krátká paréza končetin, žízeň, hlad, zvracení, pomočování a defekace. Poté již nastává normální stav. [4]

2.4 Klasifikace

Navzdory tomu, že třídění epilepsie prošlo složitým vývojem, dodnes v její klasifikaci panuje nejednotnost, která je dána zejména rutinním používáním starší popisné klasifikace epileptických záchvatů a současně řadou aktuálních poznatků z oblasti genetiky, detailních morfologických a funkčních paraklinických vyšetření, které dále upřesňují charakteristiku nemoci daného jedince a vztahu záchvatu k příčině, léčitelnosti a prognóze. Všechny tyto faktory jsou zohledněny v novější klasifikaci epilepsií. Klasifikace záchvatů, vycházejí z popisu klinických projevů pozorovatelných na pacientovi bez ohledu na veškeré další souvislosti, se v současnosti jeví jako dostačující zejména pro možné nepřesnosti v rozlišení generalizovaných a sekundárně generalizovaných záchvatů a nedostatečné rozlišení některých jednoduchých a komplexních parciálních záchvatů. Pro svou jednoduchost je však dosud v praxi hojně používána.

Klasifikace popisná má určité charakteristiky obou výše uvedených klasifikací. Předpokládá, že většina přesně popsaných a definovaných záchvatů je projevem určitého specifického epileptického syndromu a neumožňuje přesnější diferenciaci pomocí EEG křivky.

Anatomická klasifikace záchvatů je využívána v předoperační diagnostice, není určena pro běžné praktické použití.

Z pohledu lokalizace dělení na fokální a generalizované není zcela logické. Pokud předpokládáme, že pravděpodobně všechny záchvaty jsou primárně ložiskové, zaleží jen na přesné lokalizaci ložiska a na jeho vztahu k neuronovým okruhům, sítím a mozkovým strukturám s jejich epileptogenními charakteristikami, a dále na našich schopnostech ložiskový počátek postřehnout.

Existuje několik mezinárodně uznávaných klasifikací pro hodnocení typu epilepsie, avšak žádná nevyjadřuje přesvědčivou snahu dělení sjednotit. Jako vhodnější způsob klasifikace se jeví **hodnocení na základě klinických projevů**, které rozlišují následující druhy epileptických záchvatů:

2.4.1 Generalizované záchvaty

Při generalizovaných záchvatech je od počátku zřejmá spoluúčast obou hemisfér. Většinou je od počátku změněno vědomí, motorické projevy jsou oboustranné, v EEG je epileptická aktivita od počátku generalizována.

Absence znamená náhlou ztrátu vědomí s areaktivitou. Oči jsou otevřené, nemění se svalový tonus, chybí motorické nebo vegetativní projevy. Při činnosti se postižený zarazí nebo může automaticky pokračovat, na příhodu si nevzpomíná. **Absence s myoklonem** je provázena rytmickými klony hlavy, víček a ramenou, které jsou popisovány jako tiky.

Tonický záchvat je charakteristický náhlý hypertonem kosterního svalstva celého trupu, hlavy i končetin. Jeho vznik je obvykle spojen s náhlým pádem a možností poranění, protože současně nastupuje i hluboká porucha vědomí. V důsledku kontrakce dýchacích svalů je pacient cyanotický.

Klonický záchvat je sérií opakovaných více či méně generalizovaných kontrakcí kosterního svalstva a často i bránice, mezi nimiž nastává krátce svalová atonie. Porucha vědomí je rovněž přítomna.

Tonicko-klonický záchvat začíná iniciálním klonem s výkřikem, následuje asi 30–40 sekund trvající tonická fáze a asi 1 minutu trvající klonická fáze. Celkové trvání je asi 2 minuty, následuje fáze generalizované atonie s hlubokým komatem, kdy fungují jen vitální funkce.

Myoklonický záchvat se projevuje rychlými svalovými záškuby o různé frekvenci. Různé podoby myoklonií se mohou různě kombinovat v rámci jednotlivých epileptických myoklonických syndromů.

Atonický záchvat je typický poklesem nebo plnou ztrátou svalového napětí. Přítomna je různě dlouhá porucha vědomí.

2.4.2 Parciální ložiskové záchvaty

Záchvaty mají klinické i na EEG v počátku ložiskové projevy a mohou nebo nemusí být provázeny poruchou vědomí.

Simplexní parciální záchvat motorický je charakterizován ložiskovou motorickou aktivitou.

Simplexní parciální záchvat sensitivní a senzorický může mít projevy somatosenzorické, zrakové, čichové, sluchové, chuťové či vestibulární.

Simplexní parciální záchvaty s vegetativními projevy se prezentují změnami barvy, pocením a poruchami zornicových reakcí.

Simplexní parciální záchvat s psychickými projevy se může projevit poruchou amnestických funkcí, delíriem různého typu, halucinacemi.

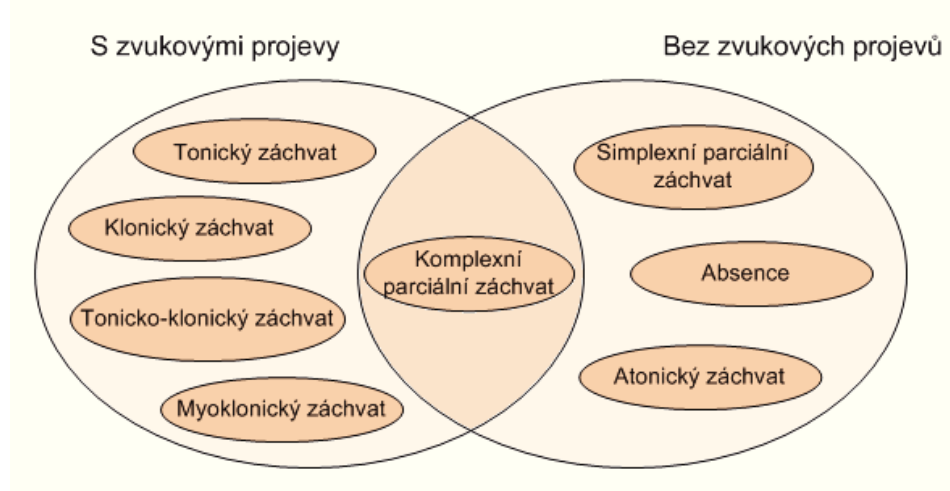
Komplexní parciální záchvat je typický automatismy, motorickými projevy, od počátku nebo v průběhu většinou nastává porucha vědomí a následuje amnézie.

2.4.3 Status epilepticus

Může se vztahovat ke všem zmíněným epileptickým záchvatům. Jde o nakupení řady uvedených záchvatů, mezi nimiž se nemocný neprobírá k vědomí. Podle mezinárodní klasifikace je definován jako záchvat trvající více než 30 minut. [3]

2.5 Identifikace na základě zvukových projevů

Některé typy záchvatů z výše zmíněných jsou provázeny svalovými křečemi různých partií pacientova těla, např. drkotání zuby, třepotem, mlaskáním apod., pomocí detekce těchto zvuků jsme schopni s určitou pravděpodobností detekovat odehrávající se záchvaty určité skupiny.



Obrázek 4.: Identifikace záchvatu na základě zvuku

3 Android

S příchodem mobilních telefonů nastala obrovská revoluce spočívající zejména v usnadnění předávání informací. Zpočátku byly tyto přístroje poměrně velkých rozměrů, vybaveny malým nepřehledným černo-bílým displejem a klávesnicí. Nabízeli uživatelům pouze strohé funkce, které se omezovaly na uskutečnění hovoru nebo zasílání textových zpráv, avšak postupem času se technologie čím dál více zdokonalovala. Došlo jednak k jejich miniaturizaci a inovaci komponent, což je hardwarová záležitost, a ruku v ruce s tím se odehrály změny na úrovni softwaru.

Software byl původně navržen na míru danému přístroji, čímž byl podstatně limitován. Oproti tomu v dnešních telefonech vládne operační systém, který organizuje spolupráci mezi hardwarovými komponentami a ostatním podružným softwarem, díky tomu se z telefonu stal multifunkční přístroj. Na poli smartphonů je momentálně jedním z dominantních OS platforma Android.



Obrázek 5.: Zastaralý a moderní mobilní telefon

3.1 Historie OS Android

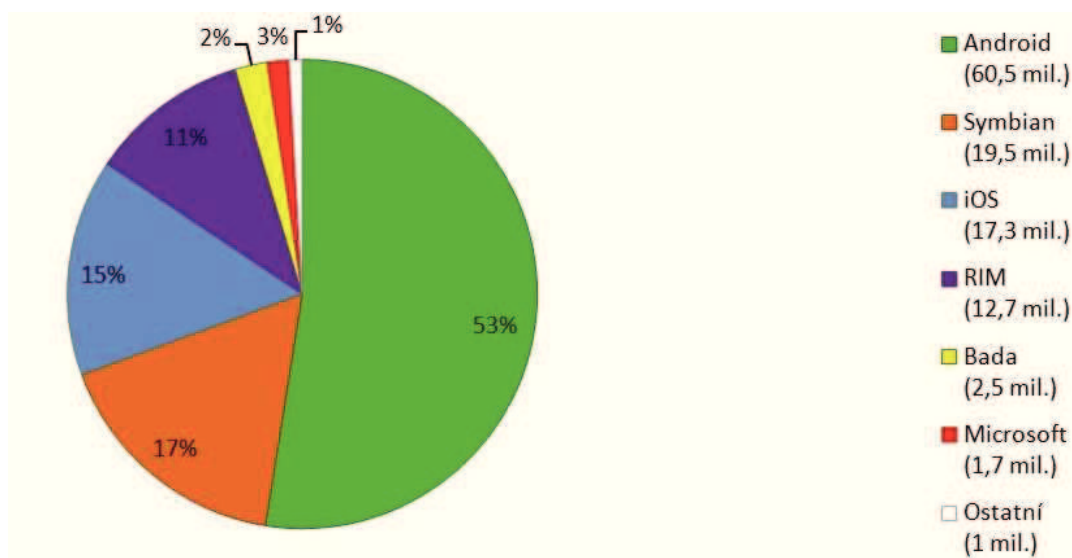
Jedná se o rozsáhlou open source platformu, která vznikla pro mobilní zařízení zejména v odvětví smartphonů. První zmínka o Androidu se objevila v roce 2003, kdy byla v Kalifornii založena společnost Android, o dva roky později o ni projevila zájem společnost Google a následně ji odkoupila. Vývojáři společnosti Google se pustili do práce a začali postupně vyvíjet platformu založenou na Linuxovém jádře. Při jejím vývoji brali v úvahu omezení, kterými disponují klasické mobilní zařízení jako výdrž baterie, menší výkonost a málo dostupné paměti.

5. listopadu 2007 Google oficiálně ohlásil platformu Android, při této příležitosti vzniklo sdružení firem Open Handset Alliance (OHA), které funguje pod záštitou Google. OHA zahrnuje 34 výrobců zabývajících se vývojem mobilních technologií. Téhož dne Google předal

platformu Android do rukou sdružení OHA. O týden později byl vydán první Android SDK pro vývojáře pod licencí open-source. Další historický okamžik se odehrál v říjnu roku 2008, kdy byl ve Spojených státech poprvé uveden komerční telefon vyrobený HTC s operačním systémem Android. Od první verze bylo vydáno několik aktualizací, které opravují chyby a přidávají nové funkce. [4][5]

3.2 Android a jeho konkurenti

Podle průzkumu provedeného analytickou společností Gartner bylo zjištěno, že 52,5% smartphonů prodaných ve třetím čtvrtletí roku 2011 bylo vybaveno OS Android. To znamená zdvojnásobení podílu na trhu oproti stejnému období loňského roku, kdy to bylo 25,3%. Celkově se prodalo přes 60 milionů zařízení s logem opatřeným zeleným robotem, což je výrazný nárůst proti 20 milionům před rokem. Android rostl na úkor ostatních operačních systémů a všem s výjimkou Bady od Samsungu klesl tržní podíl. Nejvíce se to projevilo na dosluhujícím Symbianu, který ztratil téměř 20% trhu. Současné rozložení sil konkrétních OS je výstižně zobrazeno na koláčovém grafu. [6]



Obrázek 6.: Množství prodaných zařízení za třetí čtvrtletí r. 2011 [6]

3.3 Architektura

Základní vrstva OS Android je tvořena Linuxovým open-source jádrem ve verzi 2.6, které obsahuje skupinu ovladačů nutných ke spolupráci s hardwarem a zároveň přebírá mnoho Linuxových vlastností, jako podporu správy paměti, správu sítí nebo správy procesů. Hlavním důvodem použití Linuxového jádra je výhoda poměrně jednoduchého sestavení na různých zařízeních.



Obrázek 7.: Architektura OS Android [7]

Druhá vrstva s názvem Libraries je složena se systémových knihoven napsaných v jazyce C/C++, přeložených v nativním kódu CPU přístroje. Patří sem zejména standartní knihovny jazyka C (libc), knihovny pro práci s multimédií (Media Framework), grafické knihovny (OpenGL ES, SGL, Surface manager), knihovna pro vykreslování (FreeType), knihovna síťového šifrování (SSL) a knihovna webového prohlížeče (WebKit). Funkce obsažené v knihovnách jsou vývojářům přístupné skrze Application Framework.

Android Runtime vrstva obsahuje základní knihovny programovacího jazyka Java a aplikační virtuální stroj zvaný Dalvik Virtual Machine (DVM), který je modifikací Java Virtual Machine a zajišťuje nezávislost spuštěných aplikací na hardwaru.

Dostáváme se k vrstvě pro vývojáře nejdůležitější a to je Application Framework. Zde jsou umístěny veškeré potřebné třídy a rozhraní pro tvorbu aplikací na vyšší platformě nezávislé úrovni. Aplikační Framework je napsán v jazyce Java přeložený pomocí DVM.

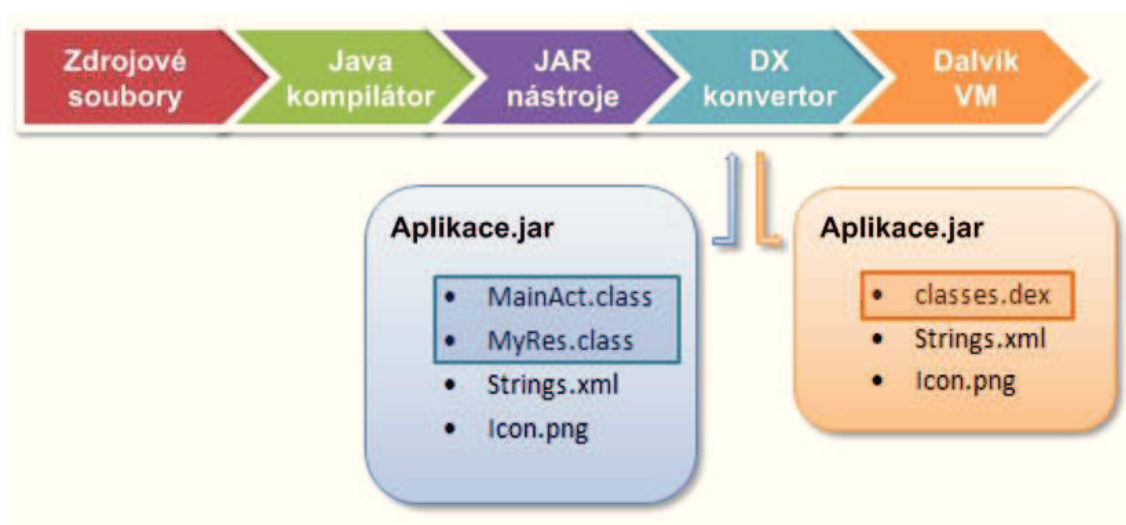
Nejvyšší vrstvu systému tvoří základní aplikace, které využívají běžní uživatelé. Může jít o aplikace předinstalované nebo dodatečně stažené z Android Marketu. [5]

3.4 Virtuální stroj Dalvik VM

Pro vývoj aplikaci na platformě Android se používá programovací jazyk Java. Základním principem Javy je přenositelnost na různé platformy. Na rozdíl od běžných programovacích jazyků, které kompilují zdrojový kód přímou metodou do strojových instrukcí, v Javě dochází k jeho částečné kompilaci na tzv. mezikód (bytecode). Při spuštění programu je bytecode přeložen na virtuálním stroji do strojového kódu a následně vykonán.

V případě Javy je zkompilovaný zdrojový kód uložen do souboru s příponou *.class. Pokud je těchto souborů více sdružují se v souboru *.jar (Java archive). Během vykonávání programu je většinou použito několik tříd a právě spouštění jednotlivých tříd v JVM je velice náročné vzhledem k omezenému hardwarovému vybavení mobilních telefonů, což kompenzuje Dalvik VM. Aby byl bytecode pro Dalvik VM srozumitelný dochází po vygenerování souboru *.class k jeho převodu na soubor typu *.dex (Dalvik executable), který obsahuje několik tříd najednou, tím pádem dochází k odstranění časové prodlevy během interpretace bytecode. Každá spuštěná Android aplikace běží ve svém vlastním procesu, s vlastní instancí DVM.

[8][9]



Obrázek 8.: Vznik souboru *.dex ze zdrojového kódu

3.5 Tvorba aplikací

Pro tvorbu aplikací můžeme využít oficiálně podporované vývojové prostředí Eclipse, do kterého stačí doinstalovat rozšíření pro Android. Pokud preferujeme práci v jiném IDE (*integrated development environment*) než Eclipse, použijeme pro tvorbu a odladění aplikace nástroje SDK.

3.5.1 Android SDK

SDK můžeme stáhnout v několika verzích podle rozsahu balíčku. SDK obsahuje nástroje pro kompilaci, knihovny, dokumentaci a emulátor telefonu.

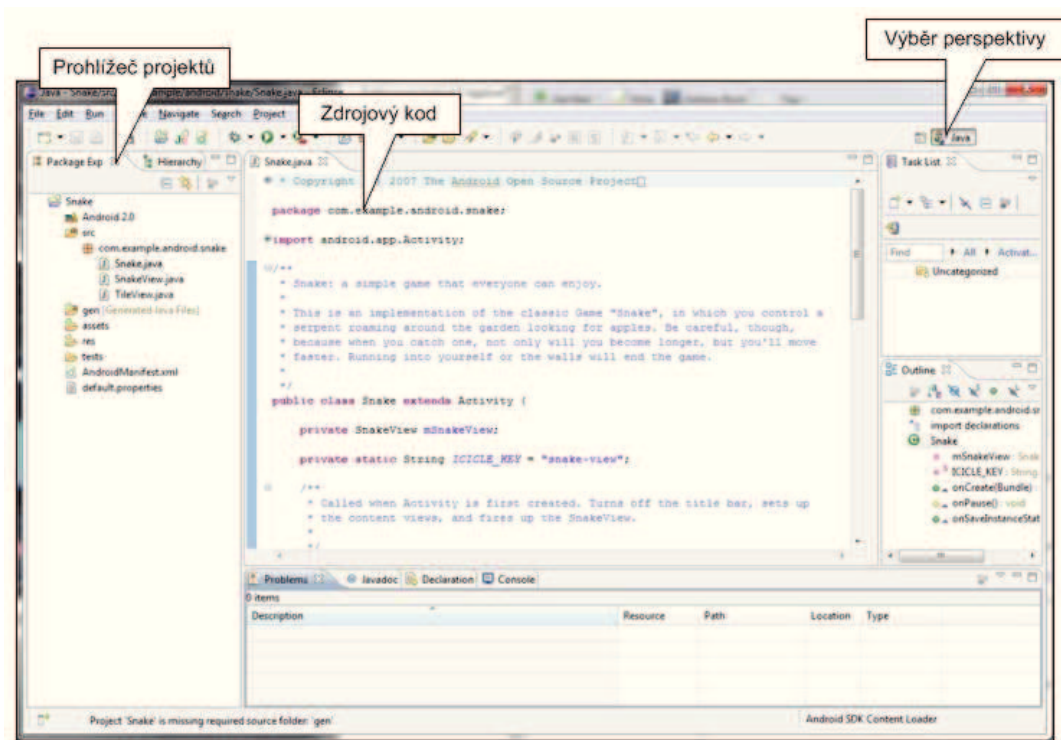


Obrázek 9.: Emulátor pro Android

Emulátor telefonu umožňuje simulovat plnohodnotné prostředí všech dostupných verzí Androidu. Díky tomu můžeme aplikaci vyvíjet a odladit na různých konfiguracích přístroje. Mezi, které patří například různé rozlišení obrazovky, velikost operační paměti případně i hardwarové zařízení jako GPS, dokonce je možnost vyvolat akce simulující příchod SMS nebo příchodí hovor. Nevýhodou emulátoru je jeho rychlost, která je na první pohled znatelně pomalejší oproti fyzickému zařízení. Proto pokud máme k dispozici fyzické zařízení, je vhodné aplikaci odladit přímo na něm pomocí USB kabelu.

3.5.2 Vývojové prostředí Eclipse

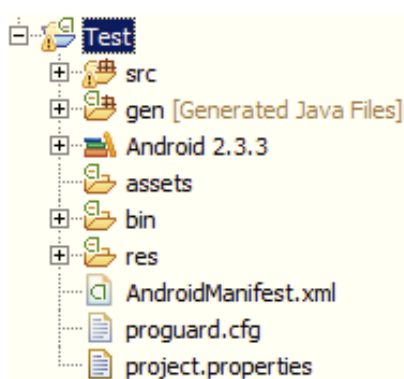
Prostředí Eclipse je rozděleno do tzv. perspektiv, což v podstatě znamená rozložení jednotlivých prvků prostředí. Perspektivy jsou pro každý programovací jazyk různé. Konkrétně u Androidu, se během vývoje používá Java perspective a pro ladění Debug perspective.



Obrázek 10.: Vývojové prostředí Eclipse rozšířeno o Android

Po instalaci rozšíření Androidu do Eclipse, se objeví v nabídce vytvořit projekt navíc možnost Android project. Jelikož Android byl vydán v několika verzích a neustále se vyvíjí, musíme si zvolit verzi, pro kterou hodláme vyvíjet a jednorázově stáhnout její aplikační Framework pokud se tak již nestalo.

Jakmile založíme projekt, udělá za nás Eclipse kus práce a automaticky vygeneruje základní adresářové struktury. Takový to projekt můžeme spustit jako funkční aplikaci.



Obrázek 11.: Struktura adresáře projektu

- **adresář src-** obsahuje zdrojové soubory
- **adresář gen-** adresář s vygenerovanou třídou R.java obsahující identifikátory prostředků (resources), která by se neměla ručně měnit
- **přilinkovaná jar knihovna build targetu-** v našem případě Android 2.3.3.
- **adresář assets-** slouží k umístění různých datových souborů, které bude aplikace využívat
- **adresář res-** obsahuje prostředky aplikace
- **AndroidManifest.xml-** popisuje aplikaci a její komponenty
- **default.properties-** slouží pro úpravu properties používaných programem ant při sestavování aplikace
- **proguard.cfg-** tento soubor se objevil nově s Androidem 2.3, lze jím definovat, jakým způsobem proběhne optimalizace a zamíchání výsledného kódu

[10]

3.6 Základní části aplikace

Základní stavební kameny v aplikacích Android jsou komponenty:

- **Activity** reprezentující obrazovku
- **Service** umožňující provádět akce na pozadí
- **Content providers** poskytující přístup k datům
- **Broadcast receiver** reagující na příchozí oznámení

Všechny tyto komponenty musí být definovány v souboru AndroidManifest.xml, uloženém v kořenovém adresáři projektu. Komponenty mohou mezi sebou kooperovat pomocí zpráv, tzv. Intentů kromě Content provideru.

[5]

3.6.1 Activity

Activita obsahuje grafické uživatelské rozhraní pro interakci s uživatelem a odpovídá jedné obrazovce. Většina aplikací obsahuje více než jednu aktivitu, mezi kterými je uživatel schopen přepínat a přitom si activity mohou předávat informace. Za správu životního cyklu activity zodpovídá Activity Manager, čímž se zabrání zbytečným plýtváním výpočetních prostředků. Activity Manager pracuje se zásobníkem, ve kterém jsou uchovány informace o spuštěných aktivitách. Životní cyklus activity se může nacházet v těchto stavech:

- **Activity starts-** Počátek, dochází k inicializaci activity.
- **Activity is running-** Activity je zobrazeno na displeji a může mít interakci s uživatelem.
- **Process is killed-** Zrušení activity kvůli nedostatku paměti.
- **Activity is shut down-** Ukončení activity.

3.6.2 Service

Komponenta service neposkytuje uživatelské rozhraní, ale představuje pouze proces běžící na pozadí. Většinou se používá k vykonávání dlouho trvajících úkolů nebo k přístupu k vzdáleným zdrojům, kde není známá doba odezvy.

3.6.3 Content provider

Content provider je aplikační rozhraní pro sdílení dat mezi aplikacemi, ale i pro sdílení dat v aplikaci mezi jednotlivými aktivitami. Aplikace může uchovávat data v souborech, SQLite databázi nebo na webu, a přesto budou mít k těmto datům přístup jiné aplikace.

3.6.4 Broadcast receiver

Broadcast receiver je komponenta sloužící k „naslouchání“ oznámení. Podle určení na ně reaguje, například výpisem na stavový řádek nebo spuštěním jiné komponenty. Příklad použití může být reakce na oznámení o nízkém stavu baterie.

[5]

4 Návrh grafického uživatelského rozhraní

Grafické uživatelské rozhraní (GUI) je optimalizováno pro mobilní telefony, je navrženo tak aby bylo možné efektivně pracovat se zařízením pomocí dotykového displeje. Vlastní GUI vytvoříme velice snadno buď napsáním příslušných příkazů přímo do zdrojového kódu, nebo deklaraci vzhledu GUI do XML souborů (layoutů).

Nastavení grafického uživatelského rozhraní včetně jednotlivých layoutů jsou obvykle uloženy ve zdrojích (resources).

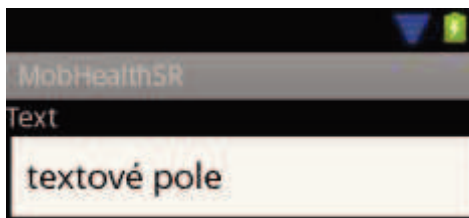
4.1 Prvky GUI

Mezi nejčastěji používané prvky patří text (TextView), textové pole (Edit Text), tlačítko (Button) atd.

4.1.1 TextView a Edit Text

TextView je prvek patřící do třídy `android.widget.TextView` a používá se jako textový popis. Obsah popisku je uložen ve vlastnosti `android:text` a lze je formátovat.

Prvek `EditText` umístěn ve třídě `android.widget.EditText`, slouží ke vstupu textových hodnot.



Obrázek 12.: Text a textové pole

4.1.2 Button

Nejdůležitější prvek, který vykonává nějakou akci, je tlačítko, patří do třídy `android.widget.Button`. Oživit tlačítko dokážeme pomocí tzv. listenerů, které poslouchají události a reagují na ně. Nejpoužívanější reakci na tlačítko je klik.



Obrázek 13.: Tlačítko

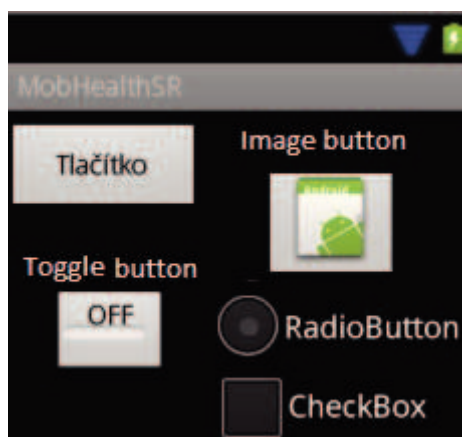
```

Button = (Button)
findViewById(R.id.bButton);
mRecordButton.setOnClickListener(new
OnClickListener() {
    public void onClick(View v) {
        //proved kód
    }
});

```

Obrázek 14.: Zápis listeneru reakce na klik

Listener z Obr. 14. lze uplatnit na většině prvků. Kromě tohoto existují ještě další typy listenerů používané pro konkrétní případy.



Obrázek 15.: Varianty tlačítka

Další varianty tlačítka jsou zobrazeny na Obr.15.

- `Android.widget.ImageButton` – tlačítko s námi definovaným obrázkem
- `Android.widget.ToggleButton` – přepínač s světelnou indikací
- `Android.widget.RadioButton` – dvoustavové tlačítko
- `Android.widget.CheckBox` – dvoustavové zatrhávací tlačítko

4.2 Rozvržení prvků

K pozicování jednotlivých prvků GUI na displeji slouží layouty.

4.2.1 Linear layout

Tento layout je vhodný zejména na učení, protože je relativně přímočarý. Jednoduše všechny do něj vložené prvky srovná do řady a to buď vertikálně, nebo horizontálně. Vzhledem k tomu, že layouty lze vkládat do sebe, dá se i s tímto jednoduchým prostředkem dosáhnout zajímavých výsledků.

4.2.2 Relative layout

Vložené prvky pozicuje relativně proti sobě. U prvku je tedy definováno, že je nad, pod nebo vedle nějakého jiného prvku. Typicky se RelativeLayout používá pro základní rozvržení obrazovky, ostatní layouty pak pro pozicování jednotlivých skupin prvků.

4.2.3 ListView

ListView je specialista na seznamy. Zobrazuje jednotlivé řádky, umožňuje jejich rolování, postupné načítání a podobné vymoženosti.

4.2.4 Ostatní layouty

Další layouty již nemají tak široké použití, nabízejí například rozložení prvků do tabulky (TableLayout) nebo práci se záložkami (TabLayout).

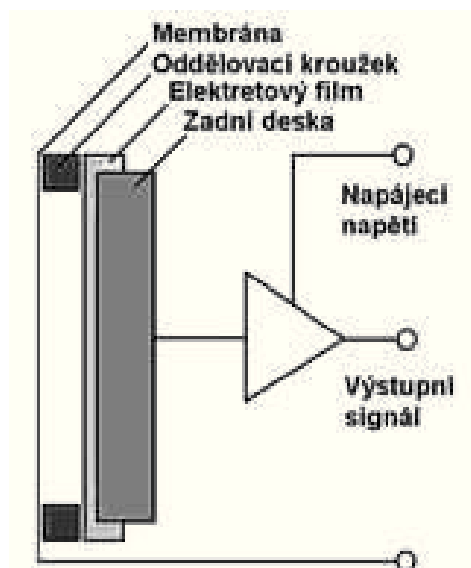
[12]

5 Zpracování zvuku

Podle fyzikální definice je zvuk mechanická energie vznikající rozkmitáním hmotných částic, které se šíří od zdroje zvuku za pomoci vazebných sil mezi částicemi prostředí. Tuto energii dokážeme zachytit mikrofonom, což je akusticko-elektrický měnič. V mobilních telefonech se používají elektretové mikrofony.

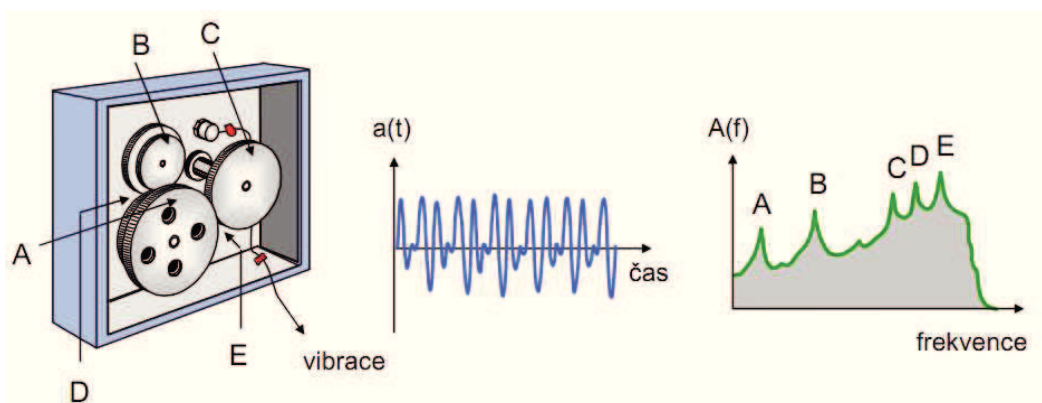
Elektretový mikrofón je typem kondenzátorového mikrofónu. Elektrický signál vzniká pohybem vodivé membrány v elektrickém poli, které je vytvářeno díky vlastnosti elektretu, tedy nevodivé hmoty permanentně elektricky nabitě. V rytmu pohybu membrány se mění velikost kapacity a zároveň napětí mezi deskami, které reprezentuje daný zvuk. Obvykle mají elektretové mikrofony FET tranzistor určený k zesílení signálu.

[16]



Obrázek 16.: Princip elektretového mikrofónu [15]

Signál získaný z mikrofónu, lze analyzovat v časové a frekvenční oblasti. V časové oblasti jsou jednotlivé jevy promíchány, zatímco ve frekvenční oblasti jsou od sebe odděleny.



Obrázek 17.: Časová a frekvenční oblast [17]

5.1 Fourierova transformace

Fourierova transformace vyjadřuje obraz signálu ve frekvenční oblasti pomocí ortogonálních bázevých souřadnic.

Vztah pro přímou Fourierovou transformaci:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt \quad (3.1)$$

,kde ω je kruhová frekvence a je definována následovně:

$$\omega = 2\pi f \quad (3.2)$$

5.1.1 Fourierova transformace diskretních signálů

K popisu diskretních signálů ve frekvenční oblasti se používá modifikace Fourierovy transformace, tzv. Fourierovy transformace diskretních (FTD) signálu obvykle označována zkratkou DTFT. Vztah pro přímou transformaci FTD se získá ze vztahu pro přímou Fourierovou transformaci (3.1) náhradou integrálu sumou. Protože je Fourierův obraz vzorkovaného signálu periodická funkce v ω s periodou $\omega_{vz} = 2\pi/T$, zapisuje se argument obrazu ve tvaru $\exp(j\omega T)$.

Vztah pro přímou FTD:

$$X(e^{j\omega T}) = \sum_{-\infty}^{+\infty} x(nT) \cdot e^{-jn\omega T} \quad (3.3)$$

V teorii diskretních signálů se často místo s kruhovou frekvencí ω pracuje s normovanou kruhovou frekvencí θ . Pomocí této normované kruhové frekvence θ vyjádřené spektrum má periodu $\theta = 2\pi$, pro FTD platí následující vztah.

$$X(e^{j\theta}) = \sum_{-\infty}^{+\infty} x(n) \cdot e^{-jn\theta} \quad (3.4)$$

5.1.2 Diskretní Fourierova transformace

Frekvenční spektrum vzorkovaného signálu $X(e^{j\theta})$ je spojitá funkce normované kruhové frekvence θ počítaná z nekonečného počtu hodnot $x(n)$. Při zpracování signálů pomocí číslicových obvodů se pracuje s konečnými počty vzorků frekvenčního spektra. Signály v oblasti času i frekvence mají stejný počet vzorků N a při výpočtech přímé i zpětné transformace se považují za periodické. Přechodem mezi signály $x(n)$ v časové oblasti a signály ve frekvenční oblasti, které se značí $X(k)$, je tzv. finitní Fourierova transformace. V případě

diskrétních signálů, kterých se tato práce týká, se nazývá diskrétní Fourierova transformace a označuje se DFT. Tato transformace je definována vztahy:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} nk}, k = 0, 1, 2, 3, \dots, N-1 \quad (3.5)$$

Vztah (3.5) můžeme rozepsat na reálnou a imaginární složku:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{2 \cdot \pi \cdot n \cdot k}{N}\right) - j \cdot \sum_{n=0}^{N-1} x(n) \cdot \sin\left(\frac{2 \cdot \pi \cdot n \cdot k}{N}\right), k = 0, 1, 2, 3, \dots, N-1 \quad (3.6)$$

Frekvence odpovídající hodnotě k:

$$f_k = k / NT \quad (3.7)$$

Čas odpovídající hodnotě n:

$$t_k = nT \quad (3.8)$$

DFT vypočte N hodnot spektra $X(k)$ z N hodnot signálu $x(n)$. Hodnoty spektra dostaneme pro diskrétní ekvidistantní hodnoty frekvencí začínající v $f = 0$ a vzdálené od sebe o hodnoty:

$$\Delta f = \frac{1}{NT} = \frac{f_{vz}}{N} \quad (3.9)$$

Množina těchto hodnot se nazývá DFT mřížka. Hodnota Δf se nazývá frekvenční bin. NT je pak celková doba odebrání signálu.

5.1.3 Rychlá Fourierova transformace

Nevýhoda DFT, spočívá v náročnosti výpočtu pro velký počet dat. Pro výpočet N hodnot je potřeba N^2 komplexních násobení a $N \cdot (N-1)$ komplexních sčítání tzn., že čas pro vykonávání výpočtu je přibližně roven času pro uskutečnění N^2 operací.

FFT jsou velmi efektivní algoritmy výpočtu DFT. FFT se používá se pro spektrální analýzu deterministických i stochastických signálů.

Základní a nepoužívanější algoritmy FFT jsou navrženy pro délku transformace $N = 2^m$, kde m je přirozené číslo. Tyto algoritmy využívají periodičnosti a symetrií komplexní exponenciály v (3.4). Tato exponenciála se nazývá otáčecí činitel (angl. twiddle factor) a označuje se WN:

$$W_N = e^{-j\frac{2\pi}{N}} \quad (3.10)$$

Algoritmy FFT pro $N = 2$ redukují počet operací pro výpočet N bodů DFT na:

$$\frac{N}{2} \log_2 N = \frac{Nm}{2} \quad (3.11)$$

Poměrná úspora v počtu operací a době výpočtu exponenciálně roste s délkou transformace. Výpočet pro $N = 1024$ proběhne zhruba 200x rychleji než při výpočtu podle definice DFT.

[13]

5.2 Vlnková transformace

Vlnková transformace je způsob jak odlišit jednotlivé komponenty, ze kterých se signál skládá a vhodně je zobrazit. Na rozdíl od Fourierovy transformace, není Vlnková transformace jedinou transformací, ale jedná se o jistý typ transformací se stejnými rysy, vzájemně se lišících podle tvaru zvolené báze funkce neboli vlnky. Od běžných transformací se liší zejména tím, že každá báze funkce je podporována pouze na konečném časovém intervalu. Následkem toho, kterákoliv hodnota spektra, využívající tuto vlnku, je ovlivněna pouze příslušným úsekem signálu. Vlnkové báze funkce pokrývají celý rozsah analyzovaného signálu po částech, tak že informace je zachována.

5.2.1 Wavelet-vlnka

Wavelety se používají k rozkladu signálů. Postupně se používají změřitkové a posunuté verze „mother wavelet“ od které jsou odvozeny. Umožňují lépe zpracovávat nepravidelné signály a signály s prudkými změnami. Díky postupnému změřitkování umožňují i prostorovou lokalizaci vlastností.

5.2.2 Spojité vlnkové transformace

Princip výpočetního algoritmu spojitě vlnkové transformace CWT spočívá ve vzájemném porovnávání analogového signálu $x(t)$ se zvoleným vzorovým tvarem, označovaným jako vlnka ψ . Vzdáleně to můžeme připodobnit, porovnávání různě tvarovaných předmětů s nějakým základním geometrickým tvarem typu krychle, kvádr, válec, kužel nebo koule. Porovnání získám hodnota CWT hodnotu koeficientu (τ, s) udávající úroveň podobnosti vzoru-vlnky se signálem.

Samotné porovnávání probíhá metodou postupného posuvu vlnky vzhledem k signálu o konstantní krok (časové posunutí τ). Výsledkem je posloupnost-řada čísel hodnotící podobnost vlnky se signálem v daný časový okamžik. Druhá možnost je, že část signálu může být vlnce podobná, pouze je časové prodloužená nebo smrsknutá.

CWT lze matematicky vyjádřit následujícím vztahem:

$$CWT(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \tau \psi * \left(\frac{t - \tau}{s} \right) dt \quad (3.12)$$

5.2.3 Diskrétní vlnkové transformace

DWT si můžeme představit jako speciálně vzorkovanou CWT, kterou lze počítat rychlým algoritmem, tvořeným filtrací FIR filtry a podvzorkováním. Jeho nejpoužívanější verzí je ekvidistantní vzorkování v čase a logaritmické vzorkování v měřítku, které nám umožňuje nestejnoměrně vzorkovat signál s ohledem na jeho časový nebo frekvenční průběh.

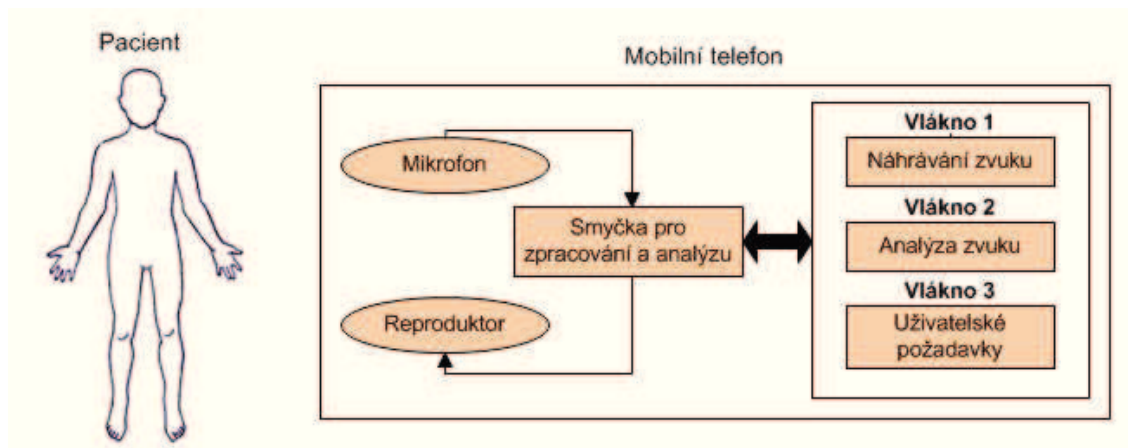
Nejčastější praktická realizace DWT spočívá ve struktuře páru kvadraturních zrcadlových filtrů tvořených dolní propustí DP (scaling filter) a horní propustí HP (wavelet filter). Základem dolní propusti je měřítková funkce a z ní vypočítaný měřítkový filtr w , z kterého normováním vzniknou koeficienty filtru dolní propusti h . Koeficienty horní propusti g pak vzniknou inverzí impulsní funkce dolní propusti h .

Výstupy obou filtrů jsou pak decimovány, tj. podvzorkovány na polovinu vstupních vzorků. Výsledkem filtrace každého stupně je posloupnost koeficientů/vzorků tzv. detailů DWT (cD) z větve s horní propustí HP a posloupnost koeficientů/vzorků tzv. aproximace (cA) z větve s dolní propustí DP. Díky decimaci je celkový počet koeficientů po jednom kroku stejný jako počet vstupních vzorků. Koeficienty aproximace lze dále analyzovat dalším stupněm banky filtrů a obdržet tak další soubor koeficientů aproximace a detailů. Tak lze postupovat až do úplného vyčerpání informace ze vstupního signálu, to znamená, že již nelze žádná další detaily nalézt.

[14]

6 Návrh aplikace pro analýzu externího signálu

Na Obr. 18. je znázorněna struktura aplikace, která se skládá s několika vláken. Ve hlavním vlákne probíhá interakce mobilního telefonu s uživatelem, ostatní vlákna pracují na pozadí a mají za úkol zpracování a následnou analýzu získaných dat na vstupu mikrofonu mobilního telefonu.



Obrázek 18.: Blokové schéma aplikace

6.1 Nahrávání zvuku

Pokud chceme využívat hardware telefonu a skloubit jeho funkci s naší aplikací, budeme potřebovat soubor procedur a knihoven, které se souhrnně nazývají API (application programming interface). Veškeré knihovny a popis jejich funkcí nalezneme na oficiálních stránkách společnosti Android.

Pro získávání dat z mikrofonu jsou využívány 3 knihovny :

- **import** android.media.AudioFormat;
- **import** android.media.AudioRecord;
- **import** android.media.MediaRecorder;

Knihovna AudioFormat se používá pro přístup k množství audio formátu a přednastaveným konfiguračním kanálům.

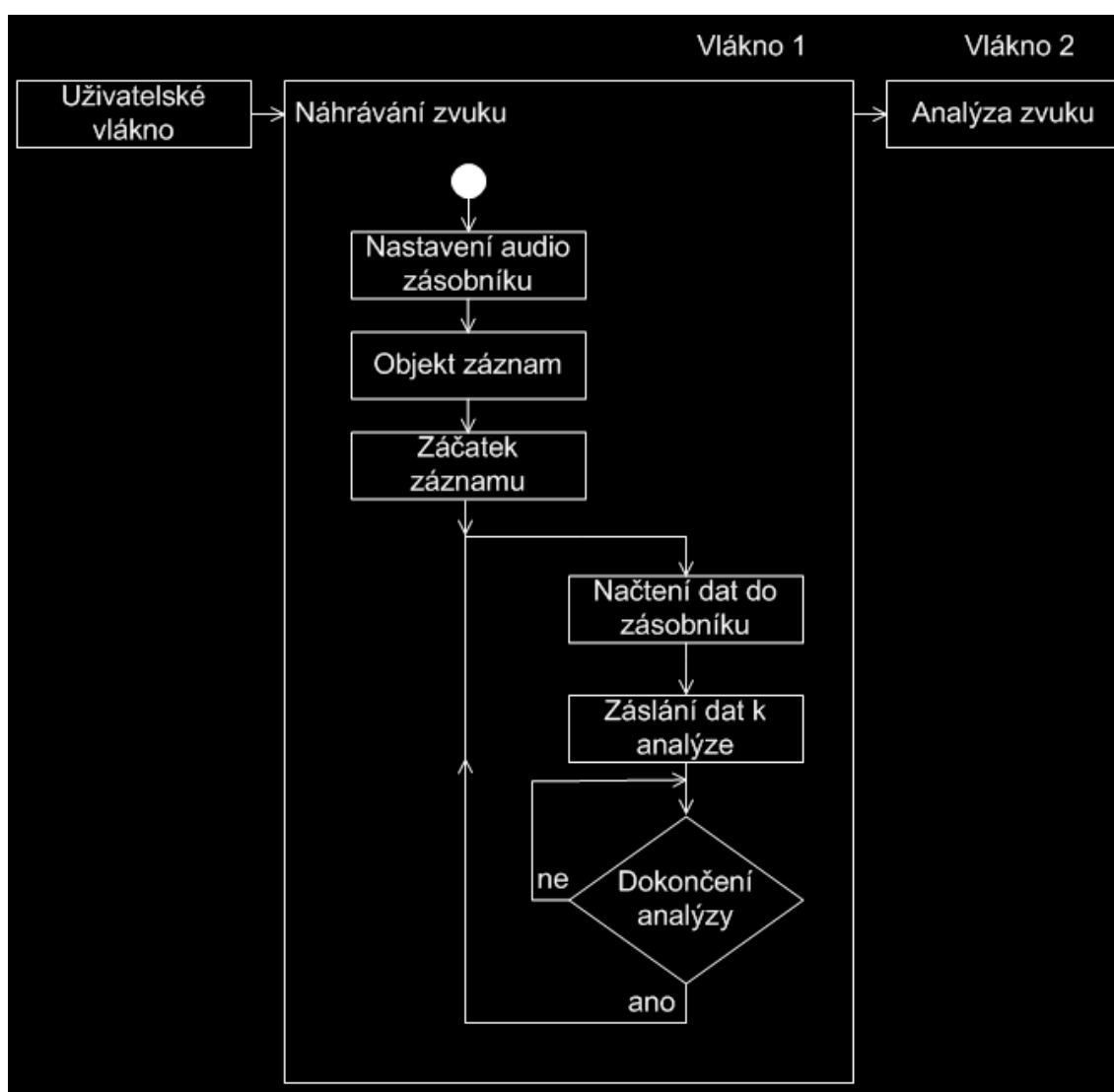
Knihovna MediaRecorder obsahuje podtřídu AudioSource, která definuje konstantu pro zařízení určené k nahrávání.

Zásadní je knihovna AudioRecord, která spravuje zvukové zdroje aplikací pro záznam z audio vstupu hardwarové platformy. Toho dosáhneme čtením dat z objektu této třídy. K vytvoření objektu této třídy nám poslouží parametry výše zmíněných tříd. Jakmile je objekt vytvořen, je možné zahájit čtení dat z mikrofonu a ukládat je do zásobníku, kde jsou data pravidelně aktualizovány.

Aby bylo ze zvuku vytěženo dostatek informací k analýze je použita vzorkovací frekvence 44100Hz, čili každou sekundu je načteno 44100 vzorků, které jsou průběžně po částech o velikosti 8192 vzorků ukládány do zásobníku, díky tomu je signál věrněji přenesen z analogové do digitální oblasti.

Než nastane situace přemazání dat v zásobníku, jsou data zaslány prostřednictvím ukazatele do dalšího vlákna ke zpracování.

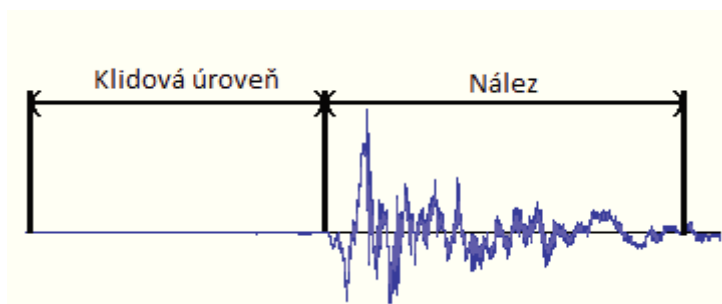
Tento děj se neustále opakuje od okamžiku spuštění vlákna, až do jeho ukončení uživatelem viz Obr. 19.



Obrázek 19.: Vývojový diagram čtení dat

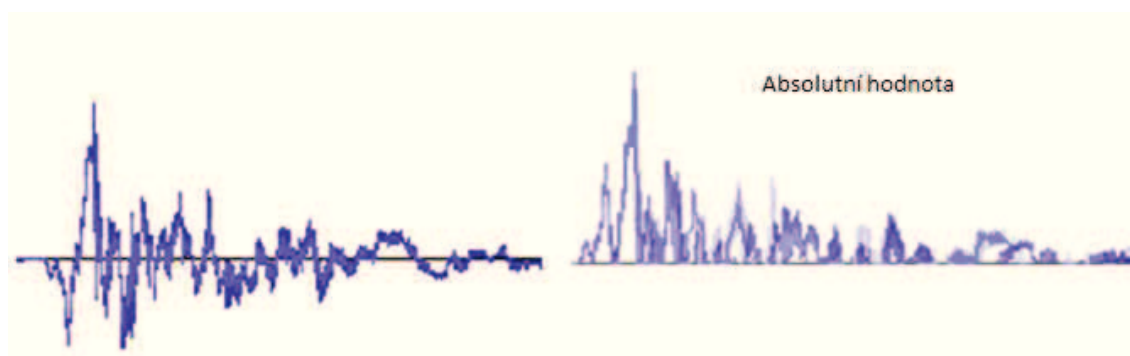
6.2 Rozpoznání zvukového nálezu

Během nahrávání dat je nutné zároveň provádět rozlišení mezi klidovou úrovní signálu, tzn. stav kdy mikrofon není schopen zachytit žádné zdroje zvuku pouze neurčitý šum a výkyvy nad tuto klidovou úroveň.



Obrázek 20.: Záznam zvukového nálezu

Na záznamu můžeme pozorovat vychýlení membrány mikrofonu oběma směry, což se v digitálním formátu projeví poklesem hodnot pod úroveň klidového stavu, z toho důvodu záporné půlky změníme na kladné pomocí absolutní hodnoty.



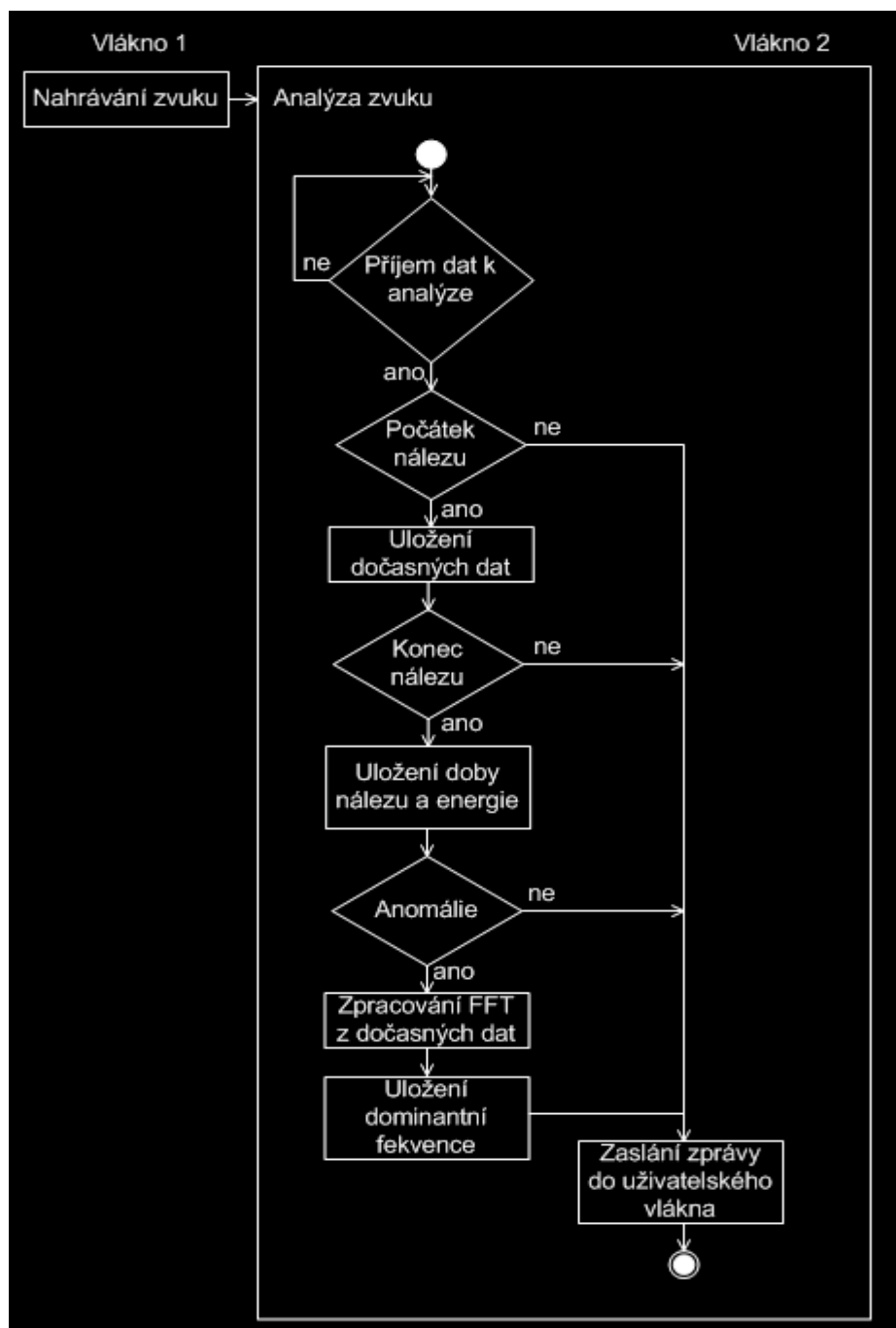
Obrázek 21.: Přepočtení nálezu pomocí absolutní hodnoty

S daty upravenými tímto způsobem dále pracujeme.

Samotné rozpoznávání nálezu je založeno na kontinuálním procházení upravených dat z audio zásobníku. Obsah zásobníku je rozdělen na několik segmentů po n vzorcích, jejíž hodnota je zprůměrovaná a následně je ověřeno podmínkou, zda má průměrná hodnota n vzorků dvojnásobnou hodnotu oproti průměrné hodnotě klidové úrovně, pokud je podmínka splněná považujeme tento okamžik za počátek nálezu.

Jakmile máme informaci o počátku nálezu, začneme data ukládat do dočasné proměnné kvůli dodatečnému zpracování FFT z obdržených dat. Za tímto účelem používáme dynamické pole. Výhoda dynamického pole spočívá v možnosti libovolné délky záznamu a navíc předejdeme vzniku výjimky v aplikaci v případě, že by data překročili rozměr statického pole.

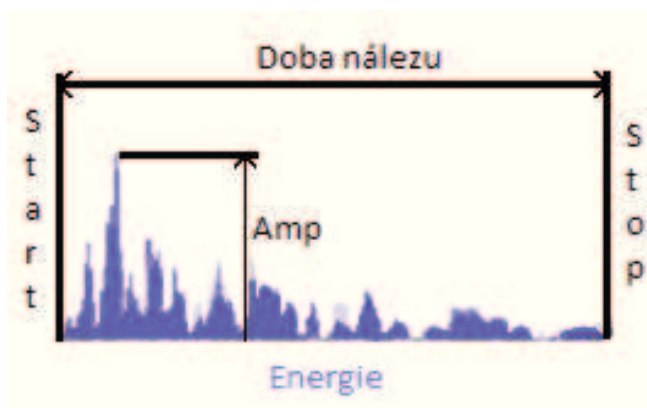
Od chvíle kdy aplikace odhalí začátek nálezu, je testována podmínka, zda nález již neskončil, která vychází z předpokladů, že hodnota zprůměrovaných n vzorků dat by neměla klesnout po 1,5 násobek zprůměrovaných n vzorků klidové úrovně. Pokud se tak stane, je systém informován o konci nálezu a opět čeká na počátek dalšího nálezu.



Obrázek 22.: Vývojový diagram detekce nálezu

6.3 Informace o nálezu

Nálezy vyskytující se v signálu můžeme hodnotit jak v časové oblasti, tak v oblasti frekvenční. V časové oblasti patří mezi podstatné parametry pro analýzu okamžik počátku nálezu, okamžik konce nálezu, max. amplituda nálezu a jeho energie.



Obrázek 23.: Veličiny nálezu v časové oblasti

Energie signálu v časové oblasti se definuje jako velikost skalárního součinu signálu se sebou samým. Jinými slovy je to integrál nebo suma kvadrátů absolutních hodnot signálu v daném čase.

[11]

Energie u spojitého signálu:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 \quad (3.13)$$

U diskrétního signálu:

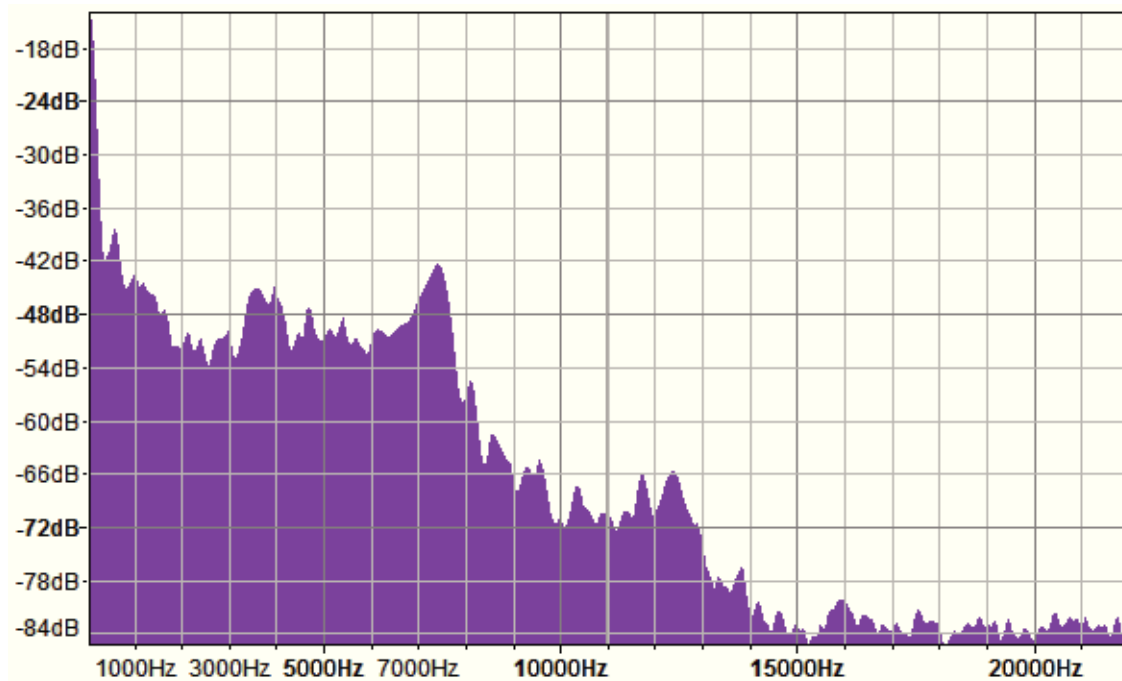
$$E = \sum_{n=0}^{N-1} |x[n]|^2 \quad (3.14)$$

Jelikož pracujeme s digitálním signálem, jehož vlastností je z principu nespojitost používáme k určení energie vzorec 3.14.

Doba nálezu definuje délku trvání nálezu a vypočteme ji z hodnot počátků a konce nálezu:

$$t_{DN} = t_{stop} - t_{start} [ms] \quad (3.15)$$

Ve frekvenční oblasti jsme schopni odlišit jednotlivé nálezy na základě podobnosti frekvence v místě maximální amplitudy frekvenčního spektra.



Obrázek 24.: FFT spektrum nálezu

```
short[] re= new short [signal.length];
short[] im= new short [signal.length];
short[] fft= new short [signal.length];
//k=0,1..N-1

for(int i=0;i<signal.length;i++)
{
    for(int k=0;k<signal.length;k++)
    {
        re[i]=(short) (signal[k]*Math.cos((2*Math.PI*i*k)/signal.length)+re[i]);
        im[i]=(short) (signal[k]*Math.sin((2*Math.PI*i*k)/signal.length)+im[i]);
    }
    fft[i]=(short) (Math.sqrt(Math.pow(re[i],2)+Math.pow(im[i],2)));
}
```

Obrázek 25.: Vzorec (3.6) pro výpočet DFT v jazyce Java

V první verzi programu byl výpočet frekvenčního spektra realizován pomocí přímé DFT (Obr. 25.), avšak při větším množství dat se ukázalo toto řešení jako příliš náročné na výpočetní prostředky mobilního telefonu. Z toho důvodu nahradil přímou DFT algoritmus FFT tzv. „Decimace v čase“.

Počet vstupních dat do FFT musí být vždy číslo 2^N vzhledem k principu algoritmu, kdy N představuje počet vstupních dat.

Počet bodů	Přímý výpočet	FFT	
N	Komplexní násobení N^2	Komplexní násobení $(N/2)\log_2 N$	Zrychlení
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.3
128	16384	448	36.6
256	65536	1024	64.0
512	262144	2304	113.8
1024	1048576	5120	204.8

Obrázek 26.: Výpočetní náročnost DFT a FFT [18]

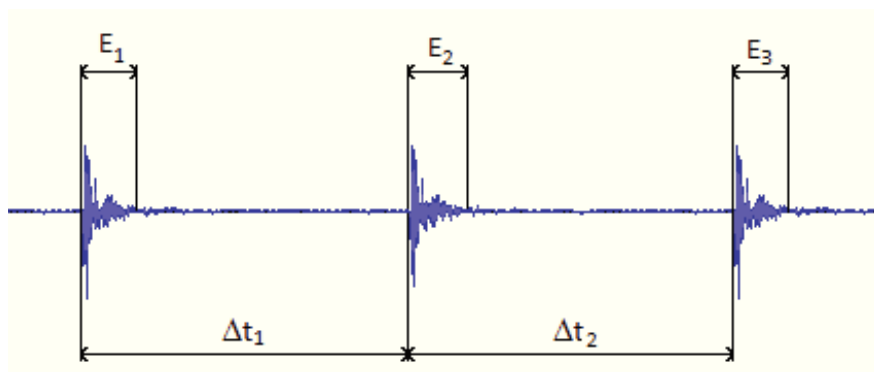
6.4 Algoritmus pro detekci epileptických záchvatů

Princip algoritmu je založen na porovnávání velikosti energií a posléze vzdálenostmi mezi jednotlivými nálezy. Jakmile je zachycen první nález dojde k archivaci jeho energie, pokud se objeví následující nález, jsou energie obou nálezu porovnávány, současně je archivovaná informace o vzdálenosti dvou výskytu.

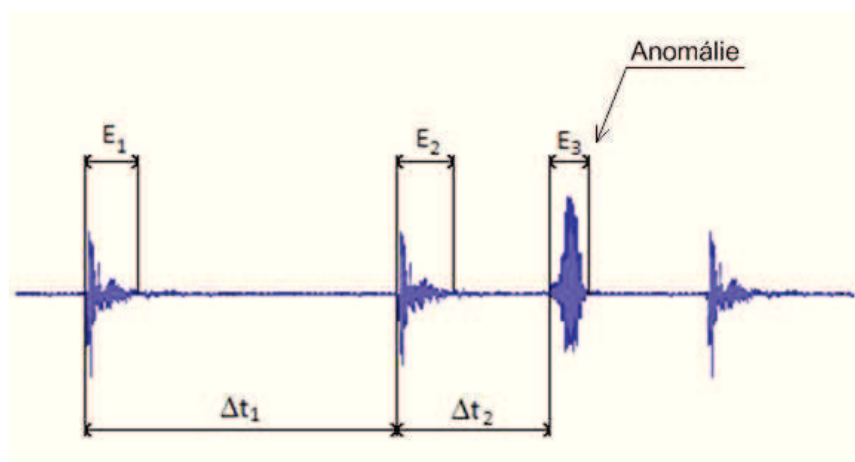
Podmínka pro určení periodických událostí a tedy nálezu, které se pravidelně opakují s určitou energií a časovou prodlevou, předpokládá že $E_2 = E_1 \pm 20\%$ a zároveň $\Delta t_2 = \Delta t_1 \pm 40\%$. Tyto odchylky byly vypořádány na základě experimentálních měření.

Na Obr. 27. je zobrazen signál splňující tyto dvě podmínky, v případě že jedna s podmínek není splněna, systém vyhodnotí nález jako anomálii a provede se výpočet FFT spektra, ze kterého získáme dominantní frekvenci nálezu.

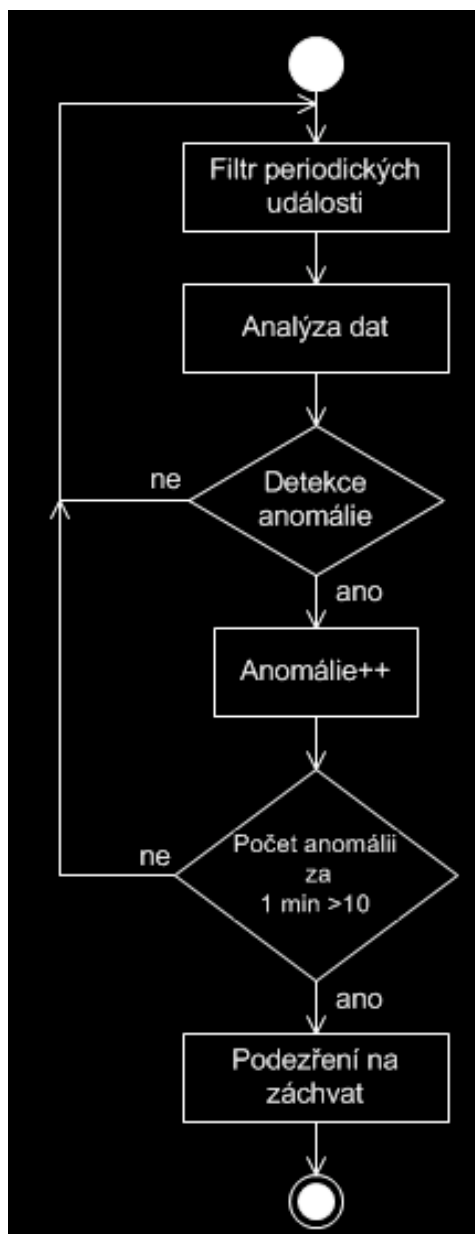
Pokud se dominantní frekvence anomálií shodují s $\pm 20\%$ odchylkou a je jich detekováno v průběhu 1 minut více než 10, je tento jev vyhodnocen jako podezření na probíhající epileptický záchvat.



Obrázek 27.: Periodický signál



Obrázek 28.: Periodický signál s anomálií

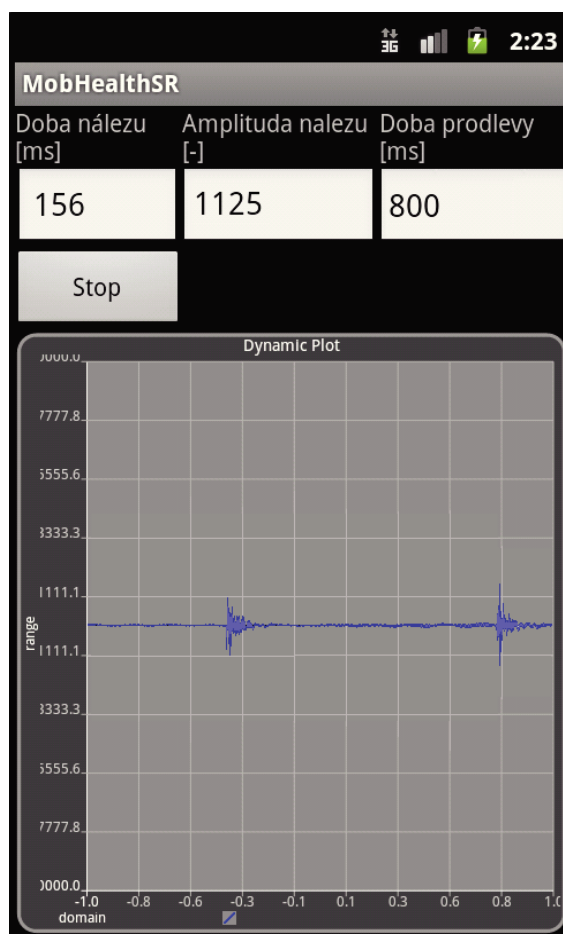


Obrázek 29.: Algoritmus detekce záchvatů

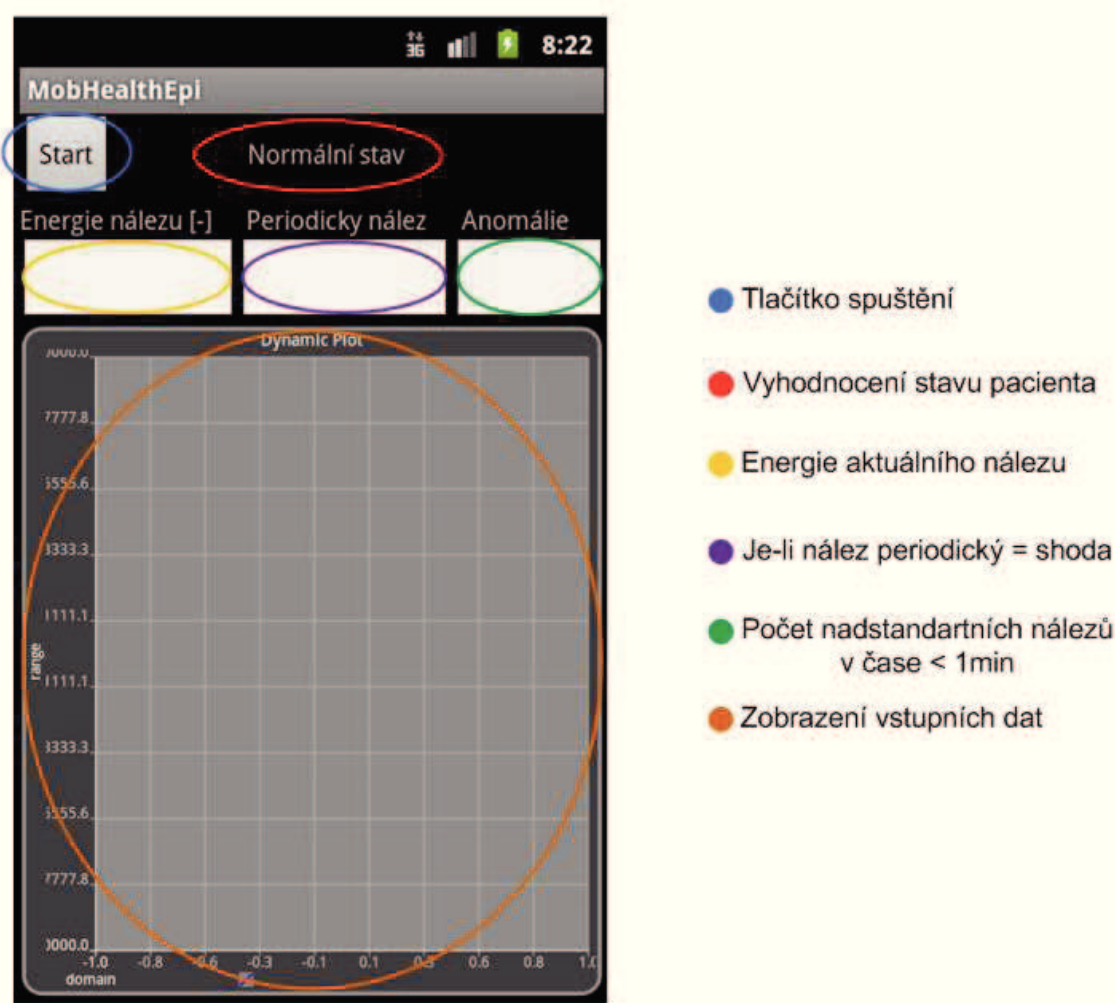
6.5 Vzhled aplikace

Aplikace pro záznam zvuku (Obr. 30.), dokáže rozpoznat zvukové nálezy a uchovávat důležité parametry nálezů, k tomuto využívá nově vytvořenou třídu `ClassifiedSample`. Velikost třídy je možné plynule měnit s ohledem na požadavky programátora. V našem případě jsou ve třídě `ClassifiedSample` podrženy parametry vždy posledního nálezu a jeho předchůdce.

Po odladění byla aplikace následně rozšířena o algoritmus sloužící k detekci probíhajícího epileptického záchvatu (Obr. 31.).



Obrázek 30.: Aplikace pro zachycení externího zvukového signálu



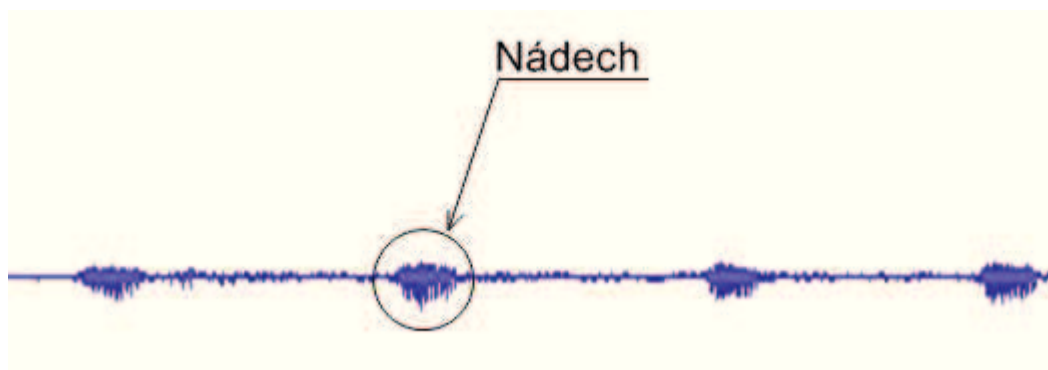
Obrázek 31.: Aplikace včetně algoritmu

K obsluhování aplikace je použito jediného tlačítka, které zahájí načítání dat. Data jsou zpracována pomocí algoritmu, jehož úkolem je rozhodnout zda skupina nálezu obsahuje množství nálezu, které mohou odpovídat epileptickému záchvatu.

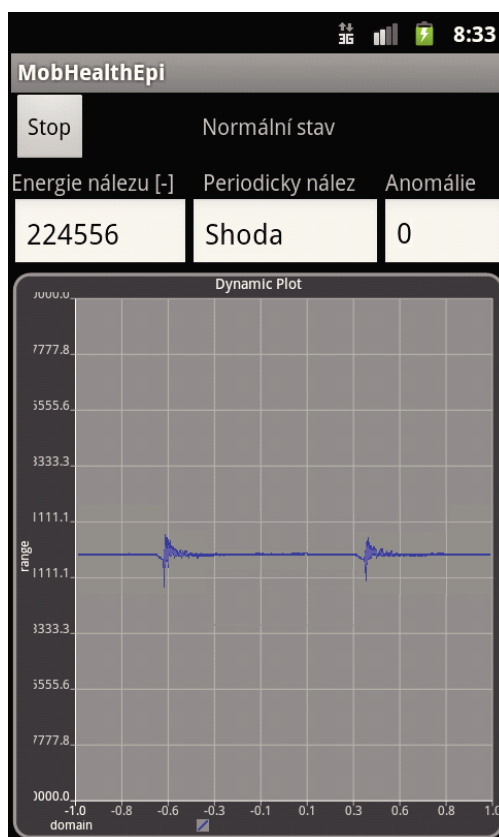
7 Testování

Ve spolupráci s FN Ostrava byly k otestování aplikace poskytnuty záznamy pacientu s diagnostikovanou epilepsií. Délka záznamu je různá, v závislosti na době po kterou byli pacienti monitorováni. Z poskytnutých záznamu byly vyňaty úseky záznamu, na kterých se vyskytují specifické zvukové nálezy a tyto úseky byly aplikovány na algoritmus.

7.1 Klidový stav pacienta



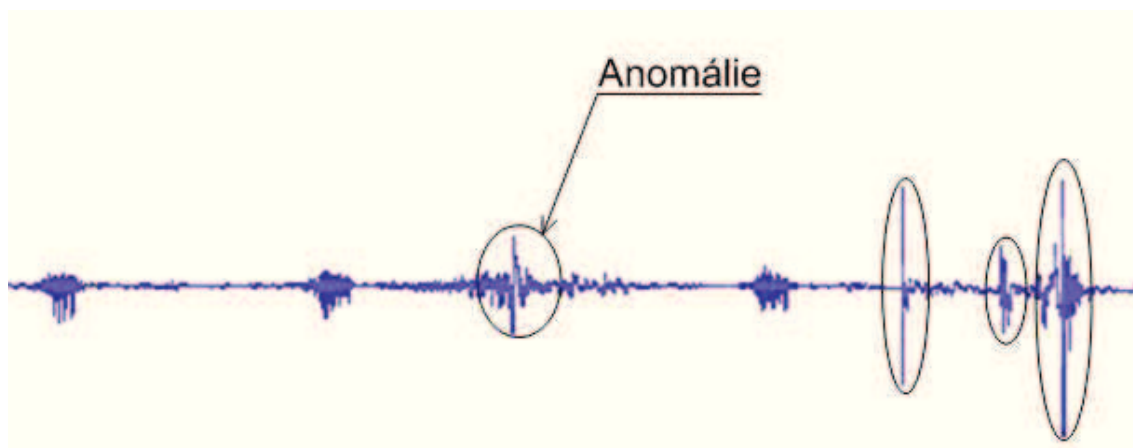
Obrázek 32.: Úsek audio stopy dýchání pacienta



Obrázek 33.: Zachycení audio stopy v aplikaci

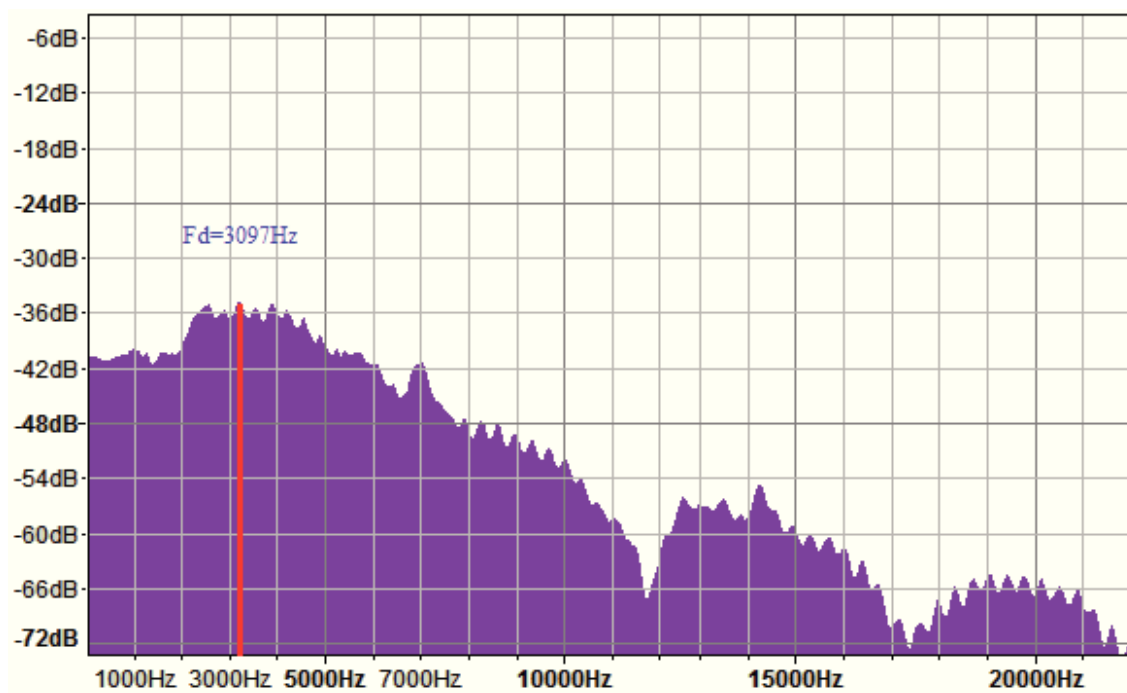
Záznam (Obr. 32.) je na první pohled periodický s podobnou energií, na základě toho byl pomocí algoritmu vyhodnocen záznam jako normální stav pacienta.

7.2 Epileptická aktivita pacienta

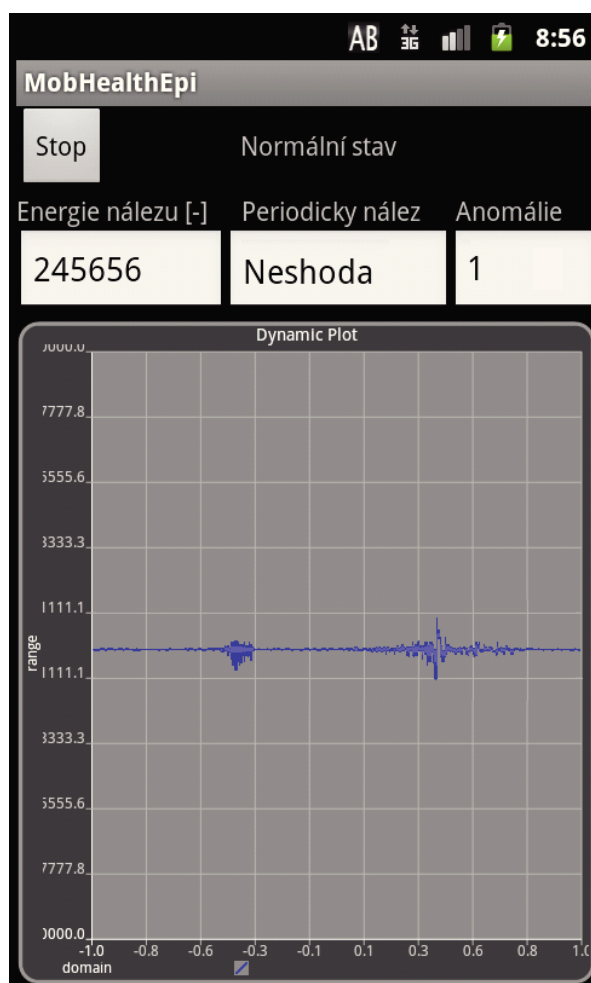


Obrázek 34.: Úsek audio stopy počátek záchvatu

V záznamu byla rozpoznána anomálie, která může být předzvěstí záchvatu. Nález anomálie je zpracován pomocí FFT. Aplikace uchovává dominantní frekvenci z frekvenční oblasti nálezů a porovnává ji s dominantní frekvencí další anomálie.

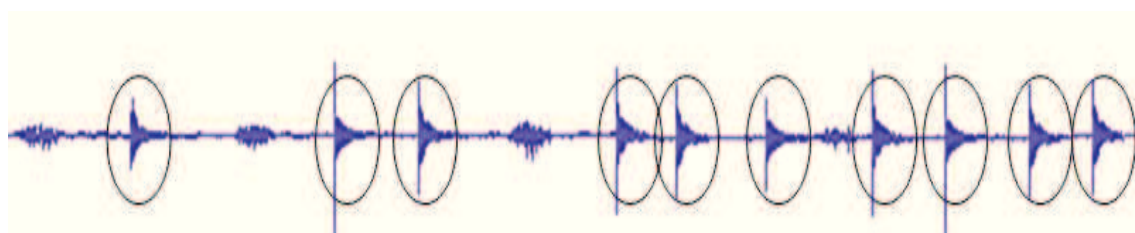


Obrázek 35.: FFT anomálie



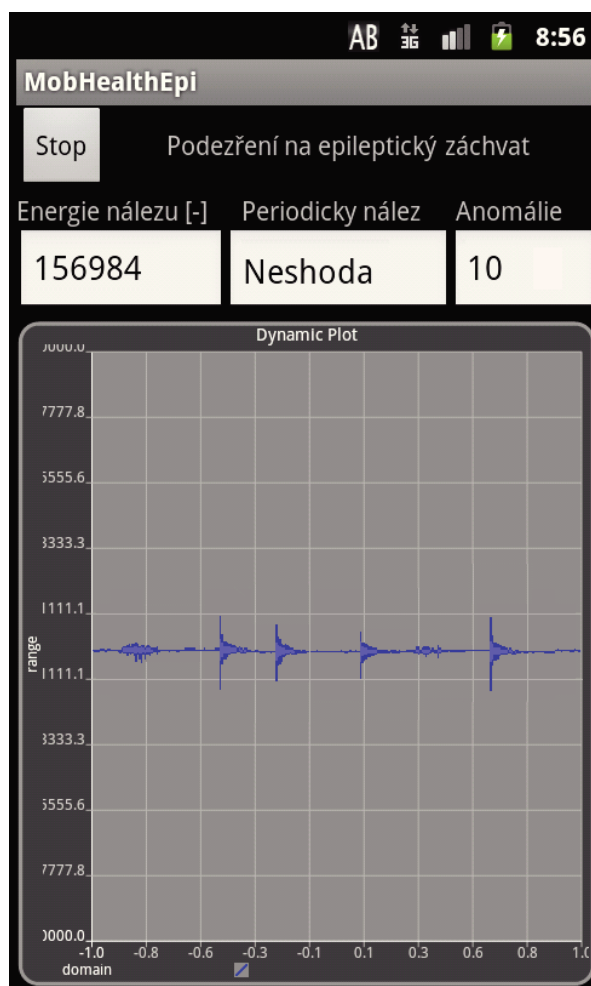
Obrázek 36.: Detekování anomálie v aplikaci

Během testování nálezu na Obr. 34. aplikace narazila na objekt, který svou periodou ani energií nezapadá do předchozích nálezu. Hodnota kolonky Anomálie je přičtena o jednu a dojde k uložení dominantní frekvence.



Obrázek 37.: Úsek audio stopy probíhajícího záchvatu

Jak můžeme pozorovat na Obr. 37. V záznamu je více nálezu, které považujeme za anomálii, vzhledem k jejich podobným dominantním frekvencím jsme určili tento nález jako probíhající epileptický záchvat viz Obr.38..



Obrázek 38.: Detekování záchvatu v aplikaci

7.3 Diskuze

Co se týče testování reálných dat, nebyly zjištěny nedostatky algoritmu, jelikož byla v našem případě jeho funkce správná. Tento fakt může být zapříčiněn skutečností malé množiny testovacích vzorků.

Pro odhalení nedostatků aplikace by se jevila jako nejvhodnější varianta, nasazení aplikace na některé s neurologických oddělení a na základě většího souboru dat provést její detailnější nastavení, případně doplnit aplikaci o vnímání dalšího vjemu, kterým by byly vibrace, snímány z akcelerometrů upevněného na lůžku pacienta, čímž bychom dosáhly spolehlivější detekce právě probíhajícího epileptického záchvatu.

8 Závěr

Cílem této práce bylo naprogramovat aplikaci pro mobilní telefony s OS Android, sloužící k detekci epileptických záchvatů na základě signálu z mikrofону přístroje. OS Android byl zvolen záměrně jednak z hlediska cenové dostupnosti přístrojů nabízených na trhu a rovněž díky hojně podpoře programových prostředků. Pro přehlednost během programování aplikace, bylo použito vývojové prostředí Eclipse. Celá aplikace byla napsána pro Android ve verzi 2.3.3. s API 10, protože jsem měl k dispozici mobilní telefon vybaven touto verzí, na kterém byla aplikace průběžně testována.

Nejdříve byla vyvinuta aplikace pro zachycení a zobrazování audio dat v reálném čase. Aplikace byla rozčleněna do 3 vláken, uživatelského vlákna, vlákna pro nahrávání a vlákna pro zvukovou analýzu, přičemž uživatelské vlákno řídí synchronizaci ostatních vláken. Jakmile byla aplikace napsaná a odzkoušena, přišla na řadu část zabývající se detekci epileptických záchvatu, za tímto účelem byl navržen algoritmus, který byl zakomponován do vlákna se zvukovou analýzou.

V posledním fázi se práce věnuje testování vytvořené aplikace rozšířené o algoritmus pro detekci epileptických záchvatu na vzorcích reálných záznamu. Podle získaných výsledku je algoritmus funkční pro námi testované záznamy, avšak abychom mohli algoritmus prohlásit za skutečně bezchybný a spolehlivě ho odladit, bylo by vhodné realizovat měření v některém z neurologických oddělení a otestovat algoritmus na větším vzorku dat.

Tento algoritmus a tato aplikace slouží jako doplňková aplikace pro kombinaci měření dat z akcelerometru zpřesňující klasifikaci epileptických záchvatů.

Literatura

- [1] Abecedazdravi. [online]. [cit. Duben 2012]. Dostupné z: <http://nemoci.abecedazdravi.cz/epilepsie>
- [2] SVATÝ, Jaromír. *Dětské nervové záchvaty*. Praha: Státní zdravotnické nakladatelství, 1964.
- [3] MORÁŇ, Miroslav. *Praktická epileptologie*. 2. vyd. Praha: Triton, 2007, 163 s. ISBN 978-807-3870-232.
- [4] Cnews. [online]. [cit. Duben 2012]. Dostupné z: <http://www.cnews.cz/google-android-velky-vylet-do-historie>
- [5] Wikipedia. [online]. [cit. Duben 2012]. Dostupné z: [http://cs.wikipedia.org/wiki/Android_\(opera%C4%8Dn%C3%AD_syst%C3%A9m\)](http://cs.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%AD_syst%C3%A9m))
- [6] Svetandroida. [online]. [cit. Duben 2012]. Dostupné z: <http://www.svetandroida.cz/kazdy-druhy-prodany-smartphone-ma-v-sobe-android-201111>
- [7] Elinux. [online]. [cit. Duben 2012]. Dostupné z: http://elinux.org/Android_Architecture
- [8] Wikipedia. [online]. [cit. Duben 2012]. Dostupné z: [http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software))
- [9] Wikipedia. [online]. [cit. Duben 2012]. Dostupné z: http://cs.wikipedia.org/wiki/Java_Virtual_Machine
- [10] Abclinuxu. [online]. [cit. Duben 2012]. Dostupné z: <http://www.abclinuxu.cz/clanky/vyvijime-pro-android-tvorime-aktivity>
- [11] Wikipedia. [online]. [cit. Duben 2012]. Dostupné z: http://cs.wikipedia.org/wiki/Energie_%28zpracov%C3%A1n%C3%AD_sign%C3%A1lu%29
- [12] Root [online]. [cit. Duben 2012]. Dostupné z: <http://www.root.cz/clanky/programovani-pro-android-zacatky-s-ui/>
- [13] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 2. přeprac. vyd. Praha: ČVUT, 2007, 255 s. ISBN 978-80-01-03110-0.
- [14] Hw [online]. [cit. Duben 2012]. Dostupné z: <http://automatizace.hw.cz/clanek/2006101401>

- [15] *Elektroakustika* [online]. [cit. Duben 2012. Dostupné z:
http://www.elektroakustika.cz/images/types_microphones/elektret2.jpg
- [16] *Technictest* [online]. [cit. Duben 2012. Dostupné z:
http://www.technictest.com/mobil_mikrofon.php
- [17] *Všb* [online]. [cit. Duben 2012]. Dostupné z:
[http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fwww.337.vsb.cz%2Fmaterialy%2Fzaklady_experimentalni_mechaniky%](http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fwww.337.vsb.cz%2Fmaterialy%2Fzaklady_experimentalni_mechaniky%2F)
- [18] MOHYLOVÁ, Jitka a Vladimír KRAJČA. *Zpracování biologických signálů*. Ostrava: Vysoká škola báňská - Technická univerzita, 2007. ISBN 978-80-248-1491-9.